Optimisation Algorithms and Heuristics for Aircraft User-Preferred Routes and their Environmental Impact

Author:
Pham, Van Viet

Publication Date:
2012

DOI:
https://doi.org/10.26190/unsworks/15671

Optimisation Algorithms and Heuristics for Aircraft User-Preferred Routes and their Environmental Impact

Viet Van Pham

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy at the
School of Engineering & Information Technology
University of New South Wales
at the Australian Defence Force Academy

Abstract

Aviation has grown dramatically over the last decades. Worldwide air traffic is expected to continue to grow at rates of 3-5% per year. Air traffic currently experiences long delays at major airports, noticeable impact on the environment, and high work load on air traffic controllers. Future operational concepts to deal with these problems and to support sustainable growth of the Air Navigation Industry have been defined by International Civil Aviation Organization (ICAO) and major Air Traffic Control (ATC) authorities including both Eurocontrol and Federal Aviation Administration (FAA). One of these concepts is User Preferred Routes (UPR), representing the routes with the best business outcome from the airspace user's perspective. Aircraft would generally be free to fly user-preferred routes, and modify their trajectories without, or with minimal, intervention by ATC. There has been limited research on optimisation methods for UPR and on investigating the impact of UPR on the environment, delays, safety, etc.

In this thesis a framework is designed and developed to find user preferred routes for flights and evaluate the UPR concept. The main contribution of the thesis is a set of three different methods for finding UPR. Through the experiments with the two first methods, which are based on Genetic Algorithms and Learning Classifier Systems and act as the black and white box optimisation approaches, we found the pros and cons for these different optimisation philosophies. The third method is a scalable algorithm that is fast and reliable. It can optimise routes efficiently for different mixes of users.

To enable the investigation and optimisation of UPR, a simulation and evaluation environment is established. In this environment, I developed a real time weather system to retrieve and process weather data. This data is then stored in a database for aviation decision support in general and for aircraft UPR in particular. Second, flight and weather scenarios are designed to test UPR methods and assess user preferred routes provided by these methods. Third, a segment based simulation environment is developed to simulate any type of route segment (climb, cruise, or descent). The simulation uses point mass model of aircraft, and is implemented in a continuous environment, as well as in a fast time mode. This simulation environment is applied effectively and flexibly to evaluate UPR.

The impact of different emissions on the environment is measured effectively. I

developed a real time flight data management system with algorithms to process, estimate, and integrate information for every flight in the airspace and to construct flight objects. Then another real time system is developed to estimate aviation emission using the flight objects provided by the flight data management system. The aviation emission is stored in 4-D database to analyse the impact of aviation emission on the environment. A number of models for emissions analysis are designed and implemented. The models developed here are then used for the calculation and analysis of fuel and emissions for UPR routes.

I demonstrate that, if UPR routes that have been optimised either vertically or horizontally, they can help to reduce on average about 3% and 2% fuel burn in comparison with the original flight plans respectively. If UPR routes are optimised both vertically and horizontally, they can save on average 5% fuel burn. I also demonstrate that the delay in departure times of UPR and non-UPR flights is insignificant (the average delay is less than 1 minute).

Keywords

Air Traffic Management, Air Traffic Simulation, Trajectory Based Operations, Trajectory Management, User Preferred Routes, User Preferred Trajectories, Aviation Meteorology, Aviation Emission, Flight Data Management, Learning Classifier Systems, Genetic Algorithms, Neural Networks, Networks, and Shortest Path Algorithms.

Acknowledgement

First of all, I would like to express my deep gratitude to my principal supervisor, Prof. Hussein Abbass, for his continuous advice, support, and supervision during my candidature. I also would like to express my sincere thank my co-supervisors Dr. Chris Lokan, and Dr. Sameer Alam for their advice, and support throughout my PhD. Particularly I would like to thank them all for teaching me research methodologies, research skills, and expertise in the different aspects in my PhD. I have learnt and grown a lot in science.

I also would like to thank Dr. Lam Thu Bui, Mr. Jiangjun Tang, Dr. Vinh Bui, and Dr. Antony Iorio for their support and collaboration.

I also would like to thank all of my fellows at the School of SEIT, the University of New South Wales at the Australian Defence Force Academy, for their assistance and their company. I particularly would like to thank Ms. Shir Li Wang, Mr. Nizami Jafarov, Mr. George Leu, Ms. Wenjing Zhao, Mr. Theam Foo Ng, Mr. Murad Hossain, Dr. Kamran Shafi, Mr. Amr Ghoneim, Ms. Shen Ren, Ms. Kun Wang, Mr. Hendra Guawan Harno, Ms. Eman Hasan, and Ms. Miriam Glennie.

Furthermore, I would like to acknowledge the financial supports from the School of SEIT. I am grateful to all the members of the school for providing such a wonderful environment.

I wish to acknowledge the contributions from colleagues at Airservices with a particular mention for Dr. Rob Porteous and Mr. Udo Bucher.

Finally, I would like to thank my family for their understanding, support, and encouragement. I feel lucky to have a family who shares with me academic enthusiasm.

Certificate of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues, with whom I have worked at UNSW or elsewhere, during my candidature, is fully acknowledged.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Viet Van Pham

Contents

List of Figures

List of Tables

List of Acronyms

AC/Eng	Aircraft/Engine
ADS-B	Automatic Dependent Surveillance Broadcast
AGA	Adaptive Genetic Algorithm
AIRMET	Airmens Meteorological Warning
AMOC	Air Traffic Flow Management Modelling Capabilities
ANSPs	Air Navigation Service Providers
AOC	Airline Operation Centers
ARFOR	Low-level Area Forecasts
ASTRA	The Australian Strategic Air TrafficManagement Group
ATC	Air Traffic Control
ATCs	Air Traffic Controllers
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
ATOMS	Air Traffic Operations and Management Simulator
ATRO	Aircraft Trajectory/Route Optimisation
ATS	Air Traffic Services
BADA	The Base of Aircraft Data
BEM2	Boeing Emission Method 2
BFS	Best First Search
BITRE	Bureau of Infrastructure, Transport and Regional Economics
BoM	Bureau of Meteorology, Australia
BRFS	Breadth first search
CAA	Civil Aviation Authorities
CDA	Continuous Descent Approach
CDL	Configuration Deviation List
CDM	Collaborative Decision Making
CDR	Conflict Detection and Resolution
CNS	Communications, Navigation, and Surveillance
CS-1	Cognitive System Level One
DFS	Depth first search
DLR	The German Aerospace Centre
DM	Decision Maker
DSGRA	Dual sequential gradient-restoration algorithm
EA	Evolutionary Algorithm

eDEP	Early Demonstration & Evaluation Platform
EDMS	Emissions and Dispersion Modeling System
ETMS	Enhanced Traffic Management System
EUROCAT	Thales Radar Data Processing Software
Eurocontrol	the European Organisation for the Safety of Air Navigation
FAA	The U.S. Federal Aviation Administration
FACET	Future ATM Concepts Evaluation Tool
FL	Flight Level
FOCs	Flight Operation Centers
GA	Genetic Algorithm
GAEC	Global Atmospheric Emissions Code
GAP	The UPR problem, UPR flights are given cruising Altitudes
GHP	Ground-Holding Problem
GPS	Global Positioning System
GRP	The UPR problem, UPR flights are given Routess
ICAO	International Civil Aviation Organization
INM	Integrated Noise Model
KB3D	A tactical conflict detection algorithm
LCS	Learning Classifier System
LTO	The Landing and Take-Off Cycle
MAGHP	Multi-Airport Ground-Holding Problem
MEL	Minimum Equipment List
MILP	mixed-integer linear program
MOEA	Multi-Objective Evolutionary Algorithm
MOP	Multi-objective Optimisation Problem
NASA	The U.S. National Aeronautics and Space Administration
NetM	Method using control point network
NetM-HV	Method using control point networks to optimise UPR routes horizontally and then verticallly
NetM-VH	Method using control point networks to optimise UPR routes vertically and then horizontally
NextGen	Next Generation Air Transportation System
NLP	Nonlinear programming
NLR	The National Aerospace Laboratory of the Netherlands
NOC	Neighboring Optimal Control
OAG	The Official Airline Guide
OPD	Optimised Profile Descent
ROC	Rate Of Climb
SAGHP	Single-Airport Ground-Holding Problem
SESAR	Single European Sky ATM Research
SIGMET	Significant Meteorological HazardWarning
SIGWX	Significant Weather Charts
SP	Singular perturbation
SSA	Shared Situational Awareness (SSA) Information Services.
SUA	Special Use Airspace

SWIM	System Wide Information Management
TAF	Terminal Aerodrome Forecast
TBO	Trajectory Based Operation
TM	Trajectory Management
TOC	Top Of Climb
TOD	Top Of Descent
TTF	Trend Type Forecast
UAV	Unmanned Aerial Vehicle
UPR	User Preferred Route
UPT	User Preferred Trajectory

List of Symbols

km	distance	kilometer
ft	distance	feet
kft	distance	kilo feet
knm	distance	kilo nautical mile
g	mass	gram
kg	mass	kilogram
kt	mass	kilotonne
s	time	second
fpm	speed	feet per minute
kN	force	kilo Newton
N	force	Newton

List of Publications

Journal Publications

1. V. V. Pham, J. Tang, S. Alam, C. Lokan, and H. A. Abbass. Aviation Emission Inventory Development and Analysis, Environmental Modelling and Software, vol 25 (12), pp. 1738-1753, ISSN: 1364-8152, Dec, 2010
2. V. Bui, V. V. Pham, A. W. Iorio, J. Tang, S. Alam, and H. A. Abbass. Bio-inspired robotics for air traffic weather information management. Transactions of the Institute of Measurement and Control, pp. 1-27, ISSN: 1364-8152, 2010

Conference Publications

1. V. V. Pham, L. Bui, S. Alam, C. Lokan and H. A. Abbass. A Pittsburgh Multi-Objective Classifier for User Preferred Trajectories and Flight Navigation, IEEE Congress on Evolutionary Computation, ISBN: 978-1-4244-6909-3, 2010.

Chapter 1

Introduction

1.1 Overview

Aviation has grown dramatically over the last decades. There are some areas in the world with a very high air traffic density (e.g US and Europe). For example, on any given day, more than 87,000 flights are in the skies of the United States, in which about one-third are commercial carriers [129]. According to an International Civil Aviation Organization (ICAO) forecast, worldwide air traffic is expected to continue to grow at rate of 5% per annum until 2020 [82].

The current air traffic systems and procedures have served the international civil aviation successfully and safely. The global accident rate remained essentially unchanged, at approximately four accidents per million scheduled departures [89].

However, air traffic currently experiences long delays at major airports [57, 30], and accounts for between 2.5% and 3% global human-made CO_2 emissions [160]. The air traffic controller workload limits the increase in the airspace capacity [55]. Future operational concepts based on the principles of Trajectory Based Operations (TBO) to deal with these problems and to support sustainable growth of the Air Navigation Industry have been defined by ICAO and major Air Traffic Control (ATC) authorities including both the European Organisation for the Safety of Air Navigation (Eurocontrol) and Federal Aviation Administration (FAA) [100, 164, 65].

These concepts are provided by SESAR (Single European Sky ATM Research) program in Eurocontrol and NextGen (NextGen Next Generation Air Transportation System) program in FAA. Trajectory-Based Operations will be based on User Preferred Route (UPR) which is the route with the best business outcome from the airspace user's perspective. User Preferred Routing will be implemented in Free Route airspace while a fixed Route Network is still maintained. Aircraft would generally be free to fly user-preferred routes, and modify their trajectories without or with minimal intervention by air traffic control [134, 78].

In trajectory-based airspace, all trajectory management functions use the aircraft's 4D trajectory with the support of digital data communication and ground-based and airborne automation. The uncertainty of an aircraft's future path in terms of predicted spatial position (latitude, longitude, and altitude) and times along points in its path is reduced dramatically as the result of using precise 4D trajectories. This enables more effective use of airspace to safely accommodate high demand and maximize the use of capacity-limited airspace and airport resources [100].

TBO in general can provide greater capacity, better efficiency, better predictability, and ground automation, improve safety, reduce fuel burn and emissions, and increase flexibility [100]. UPR in particular allow users to plan and fly optimal routes consistent with business priorities, thus minimizing flight time, mileage and/or fuel consumption, and contributing to the reduction in green house emissions [14].

This thesis addresses the UPR problem to find the best business outcome route for the airspace user. Particularly this thesis proposes a number of methods to resolve the problem. The advantages and disadvantages of the methods are investigated. A systematic use of methods is proposed to take advantage of each method in specific routing circumstances (strategic or tactical routing, and centralized or decentralized routing). The environmental impact of UPR routes, which are optimised vertically, horizontally or both for time and discomfort minimization are investigated. Here passenger discomfort is caused by bad weather and is measured by the bad weather levels and time durations a flight passes the bad weather areas. The analysis can support airspace users in making decision on their preference for a flight by weighing the environmental impact with other benefit. Besides that the impact of UPR routes on departure time deviation is also analysed. This can relatively refer to the impact of UPR routes on the delays of flights.

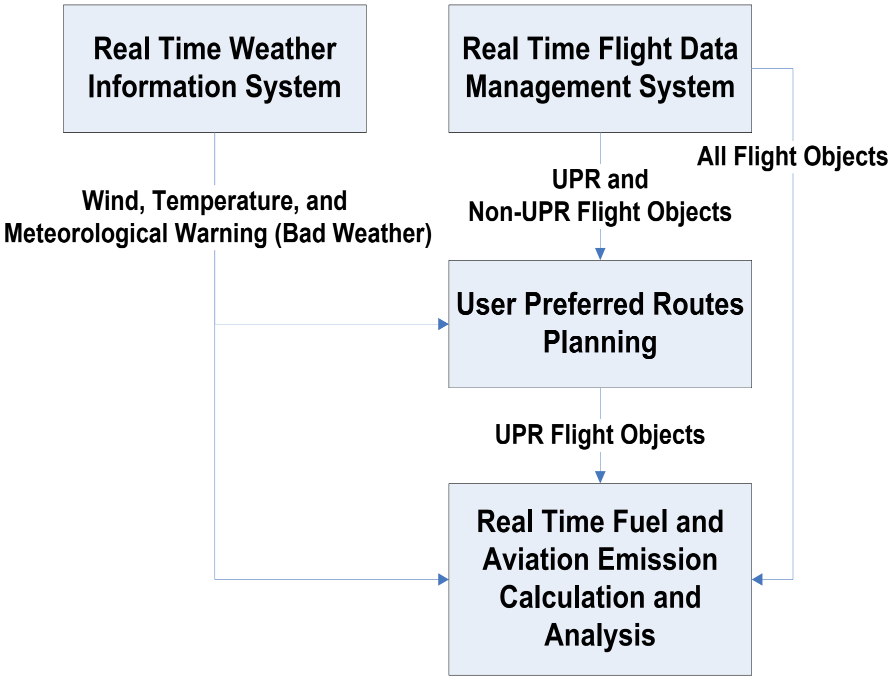


Figure 1.1: System Architecture.

In order to find user prefer routes satisfying constraints of safety and aircraft performance, and to evaluate UPR concept, a system including a number of components as in Figure 1.1 is designed and developed.

- A "Real Time Weather Information System" is designed and developed to process and integrate weather data for UPR Planning. Neural networks are designed and trained to recognize numeric atmospheric data in image format accurately and export to text format for further use. This data is also considered for use in aviation emission calculation and analysis where wind, temperature and weather effects on aviation emission are taken into account in the future.

- A "Real Time Flight Data Management System" is developed with algorithms to process, estimate, and integrate information for every flight in the airspace to construct flight objects in real time. The information includes flight plan, flight route and radar data. In the flight data management system, a number of estimation techniques are used. For example when information about runway, cruising altitude, departure, arrival routes are not received, they are estimated using statistics from historical data; some information such as actual departure times and actual arrival times are estimated through trajectory prediction. The flight objects are then used as input data for UPR planning and for calculating fuel and emission.

- "User Preferred Route (UPR) Planning" (also called User Preferred Routing) finds UPR routes for UPR flights.
- "Real Time Fuel and Aviation Emission Calculation and Analysis" estimates fuel burn and emissions in real time using flight objects provided by the flight data management system, and analyses the environmental impact of aviation emissions.

UPR planning, in this thesis, uses artificially generated flight plans because of the confidential character of the real data. The generated flights plans include UPR plans that user preferred routing needs to find user preferred routes for, and non-UPR plans that user preferred routing needs to consider to avoid conflicts with. It uses real wind data processed by the real time weather information system. However bad weather scenarios are generated artificially for testing and assessing UPR algorithms as the of quantity and distribution of bad weather cells in reality is not challenging enough for this purpose. The UPR algorithms can also work well in a real system.

In this thesis, we apply the real time fuel and aviation emission calculation and analysis system to prepare an aviation emission inventory and do analysis based on the emission data. We expect that we will be able to quantitatively determine the impact of different emissions on different flight levels, airspace areas, seasons, and weather conditions by studying aviation emission. From this, some particular environmental objectives may be defined in UPR planning. In this thesis, the models for aviation emission calculation and analysis from the system are also used to investigate the environmental impact of User Preferred Route sets.

1.2 Research Motivation

SESAR determines that 4 dimension trajectory management will be based on User Preferred Routes (UPRs), while NextGen states that Flight Operation Centers (FOCs) need to develop and maintain information technology systems to achieve the development of User Preferred Routes as one of three basic objectives. User Preferred Routes (UPRs) help to achieve the "*overall philosophy driving the delivery of Air Traffic Management (ATM) services in the NextGen to accommodate flight operator preferences to the maximum extent possible*" [100]. Restrictions are only imposed when a need for operation exists, to meet capacity, safety, security, or environmental constraints. Though UPR is described in literature, there is limited research of UPR. In [7], a continuous descent approach is discussed as an implementation of user preferred trajectories, but this is only for fuel and emission optimisation preference rather than other user preferences such as minimal discomfort. Furthermore, there has been little study to investigate the impact of UPR on the environment, delays, safety, etc. This is the motivation for studying the User Preferred Routing problem from the points of view of formulating User Preferred Problem comprehensively, developing methods to resolve the UPR problem properly and efficiently, and analysing the impact of UPR routes.

There are many methods for formulating the problem to aircraft route optimisation. Several methods work with a single aircraft only (e.g. [127]), while others focus on multiple aircraft. However they usually deal with a small number of aircraft. For example, in [148], only experiments with one, three and four aircraft were reported, while [35] reported only the case with 3 aircraft. Further, the problem is usually formulated without taking weather conditions into account such as in [127, 46, 27], without considering aircraft performance [127, 35], or in discrete environment [35], where there are a finite number of positions that aircraft can reach. This motivates us to study the problem of User Preferred Routing as aircraft route optimisation problem for large numbers of flights, taking into account user preference/intention, wind, bad weather and aircraft performance data.

UPR problem needs to find 3-D or 2-D routes accomplishing user preferred objectives (which cannot be calculated through aircraft route simulation) while satisfying constraints such as aircraft performance. These requirements create a search space with multi-dimension, multi-modal, and combinatorial-explosion. This is a difficult challenge for conventional optimisation techniques. Evolutionary Algorithms (EAs) as stochastic search methods are well suitable for this problem, because these algorithms are designed to find a good approximation rather than guaranteeing to find the best solutions. There has been a lot of research using evolutionary computation to resolve aircraft trajectory/route optimisation. These studies can be categorized into two branches. One is a black box approach, which can find optimal routes for an aircraft but gives no reasons for the aircraft movement. For example there has been a number of studies using Genetic Algorithms (GAs) [200, 9], and ant colony optimisation [4]. The main difference in these studies is in the representation and simulation of aircraft route. The other is a white box approach, which can give the reasons for aircraft navigation. There is little study in this branch. In [170], a genetics-based machine learning system is implemented to acquire rules for novel fighter combat maneuvers through simulation. In this study, a Learning Classifier System (LCS) is designed for discovering rules that dictate combat maneuver logic. There is no study using LCS to navigate civilian flights in a continuous environment, to optimise objectives such as time, fuel, emissions or comfort. There is also no study to investigate the advantages and disadvantages of these two black and white approaches. This study can support a route planner to apply algorithms properly in specific routing circumstances.

In addition to designing and developing methods based on Genetic algorithms, with an innovative and efficient representation and simulation of aircraft routes, this thesis also proposes methods based on LCS for user preferred routes for civilian flights in continuous environment to minimize time travel and maximize passenger comfort. Then the advantages and disadvantages of the two approaches are pointed out.

Another approach in aircraft trajectory/route optimisation is network based algorithms. Some methods consider pre-determined potential flight segments for finding an optimal or near-optimal route [103, 6]. Other research constructs networks for finding an optimal aircraft trajectory/route [41, 133, 128, 20, 202, 201, 111]. A network is constructed first, then a shortest path algorithm (e.g Dynamic Programming [38], A* [45], and Dijkstra [38]) is applied to find the optimal path. The main difference in these studies is in constructing the network. In this thesis we propose efficient network models for finding user preferred routes. Algorithms for determining the nodes and weights of network links to construct the networks are developed. Then Dijkstra's algorithm (with a modification to improve the speed of the algorithm) is applied to find UPR routes in the networks. This method can find UPR routes successfully satisfying the user preference in short time. Here we don't use A* which can achieve better performance (with respect to time) by using heuristics. The reason is that it is not easy or may be time consuming to have an admissible heuristic estimate of the distance from a point in airspace to the destination since the estimation needs to consider multiple objectives, and these objectives are determined in a weather environment, taking into account aircraft performance. Therefore the application of A* in this case may be more time consuming than Dijkstra.

The evaluation of UPR concept is limitedly researched. In this thesis we particularly study the environment impact of UPR concept. The aviation industry currently pays high attention on the environment impact of aviation as the high and growing volume can lead to high impact of aviation emissions on the environment. Aircraft release between 2.5% and 3% of the total global emissions of CO_2 as a result of fossil fuel burning [160]. Besides that aviation emissions have a special impact compared to the other emission sources as they are produced in high altitude. Therefore studying aviation emissions to understand their impact is important. However, the current state of knowledge of aviation impacts on the atmosphere is severely limited due to unavailability of timely and accurate aviation emission data. Most aviation emission inventories do not use real time air traffic data such as inventories developed by the U.S. National Aeronautics and Space Administration (NASA) [178], FAA [61], and Eurocontrol [59]. Several assumptions are used, to simplify complex flight trajectories and to deal with a few representative aircraft types. Moreover the emission inventories are not comprehensive i.e. they do not have details of emission in different flight phases (such as taxi-out and taxi-in). Therefore the emission data obtained are not very accurate or detailed, and by the time they are compiled they become outdated. New approaches and models are required in order to reduce the considerable uncertainties in emission calculations. This motivates us to develop a real time system for aviation emission inventory development and analysis with more accurate fuel flow and emissions computation. The availability of this real time aviation emission database will help environmental analysts and aviation experts, providing an indispensable source of information for making timely decisions regarding applying route charges based on environmentally congested airways, and restructuring air traffic flow to achieve sustainable air traffic growth, expansion of runways, building new airports. Another advantage of developing the system is that the aviation emission calculation and analysis tools can be applied to investigate the environmental impact of UPR route sets provided by different methods. This investigation is important for the implementation of UPR in the future.

In summary, the above reasons motivate us to develop a system to find UPR routes from origins (taking-off points) to destinations (landing-down points) for large numbers of flights in a weather and safety constrained airspace. The core of the system is UPR methods using evolutionary computation and networks with innovative and efficient design. UPR routes are found by these algorithms in continuous space, taking into account weather information and aircraft performance, and targeting objectives of civilian flights such as time, fuel, emissions and comfort. The performance of different algorithms is compared, then a systematic use of algorithms is proposed. A variety of weather and flight scenarios are designed to test and assess algorithms. In addition, a real time system for aviation emission inventory and analysis is developed. The environmental impacts of different sets of UPR routes, which are optimised vertically, horizontally or both for time and passenger discomfort minimization, are then analysed using the aviation emission calculation and analysis tools from the system. The departure time deviation of UPR routes is also analysed to relatively see the impact of UPR routes on the delays of flights.

1.3 Research Question And Hypothesis

Our hypothesis is that User Preferred Routes could have a significant Savings for the Aviation Industries.

As presented in Section 1.2, there is little study for finding UPR routes for flights in the literature, particularly studying the impact of UPR routes. In this thesis, we wish to answer the following research question:

What is the impact of User Preferred Routes?

This question is addressed systematically by resolving the following supportive questions.

1. **How can we effectively measure the impact of different emissions on the environment?**
   Air traffic increasingly impacts on the environment. One of the important objectives of the future concepts is to reduce the environment impact. A system to calculate aviation emissions to assess and analyse the environmental impact in real time (which includes any test of future concept) is needed. This system needs to have algorithms and models to process, estimate, predict, and integrate flight data in real time to construct flight objects. Then fuel, and emissions models can be applied to calculate fuel and emissions, using these flight objects. An efficient database to restore fuel, and emissions in

real time and analytic models are required for emission analysis. The emission calculation and analysis models in this system can be also applied for historic and simulated traffic, so that future concepts (such as UPR concept in this thesis) can be evaluated. All these problems are addressed in Chapter 4.

2. **What are the pros and cons for different optimisation philosophies?** Finding optimal solutions for a real world problem may be computationally expensive or even impossible, as the complexity of the problem prevents exact methods from being applicable. Finding an aircraft's optimal trajectory/route in continuous airspace is such a problem. Evolutionary Algorithms (EAs) are broadly used for these types of problem. These algorithms are stochastic search methods. They try to find a good approximation but do not guarantee to find optimal solutions. In this thesis we present two major sub-fields of EAs: Genetic Algorithms (GA) and Learning Classifier Systems (LCS) for solving user preferred route problem. They correspond to two different approaches: black box approach and white box approach respectively. Especially we are interested in investigating the difference (the advantages and disadvantages) between the two approaches. This can support route planners to choose a method corresponding to a routing circumstance properly. The question of "What are the pros and cons for different optimisation philosophies (black and white box approaches)?" is addressed in Chapter 5.

3. **How can we optimise routes efficiently for different mixes of users?** Real time optimisation requires fast optimisation engines to efficiently run the optimisation in a time-constrained environment. In this sub-question, an algorithm for dynamic generations of networks is used to constructively provide a fast optimisation algorithm for UPR. This question is addressed in Chapter 6.

1.4 Organization of the thesis

This thesis is organized in seven chapters as follows:

Chapter 1 presents an introduction to the thesis. The motivation and research questions are defined. Besides that, an outline of the thesis structure and the main contributions are given.

Chapter 2 reviews key concepts in the thesis. The development of air traffic management systems is introduced first, with the emphasis on future operational concepts. Then multi-objective optimisation is reviewed, presenting basic concepts of multi-objective optimisation, popular user preference elicitation techniques, and discrete and continuous optimisation algorithms (or shortest path algorithms using graph and evolutionary algorithms). Aircraft trajectory management is presented next, with technology enablers for future operational concepts, flow planning tools, and trajectory and route optimisation. User Preferred Routing is then presented, with types and nature of users in air traffic management, and variants of user preferred routes. A chapter summary and emergent questions are finally presented.

In Chapter 3, two user preferred route problems are formulated and described first. One is with flights given 2-D (latitude, longitude) routes and the other is with flights given cruising altitudes. Then a bad weather model is presented, followed by weather data processing. Conflict Detection and Resolution (CDR) algorithms used in the thesis are then presented. The simulation environment in general is presented next, followed by the mathematics of the evaluation environment and the algorithms to simulate a route. Finally the overall experimental framework to test and evaluate algorithms for User Preferred Routing is presented, followed by the summary of the chapter.

In Chapter 4, an aviation emission inventory using real time air traffic trajectory data is presented. The overview of the chapter is presented first, followed by the introduction of the chapter. Environmental impact of aviation emissions in general are presented next, followed by the description and comparison of our aviation emission inventory with other inventories. Then aviation emission results are presented in summary and for particular aircraft categories and aircraft phases. Geographical distribution of emissions in horizontal and vertical views is presented next. The discussion on the impact of aircraft emissions is given in the following section. Then limitations and assumptions in the study are presented. Recommendations for reducing the environmental impact of aviation emissions are given, followed by the summary of the chapter.

Chapter 5 presents two approaches using evolutionary computations for User Preferred Routing. One uses Genetic Algorithms and works as a black box approach and the other uses Learning Classifier Systems and works as a white box approach. The discussion and comparison of the two approaches are presented.

In Chapter 6, the design and experiment of UPR algorithms using dynamic networks of control points are presented. The overview of the chapter is presented first. UPR using a control point network for a flight, given either a 2-D route or a cruising altitude are presented. UPR algorithms using control point networks, Genetic Algorithms, and Learning Classifier Systems are compared in terms of running times and quality of solutions. The environmental impact of different sets of user preferred routes is analysed. A systematic use of UPR methods, which are based on Genetic Algorithms, Learning Classifier Systems, and control point networks is then described. Finally conclusions are presented.

Chapter 7 concludes the thesis. The major contributions of the thesis are highlighted. A number of directions for future research are pointed out.

1.5 Original Contributions

The main contribution of this thesis is to design a real time system for User Preferred Routing. Further contributions of the thesis are listed as follows:

- **A simulation and evaluation environment is designed and developed to support User Preferred Routing.**

  To enable the investigation and optimisation of UPR, a simulation and evaluation environment is established. In this environment, I developed a real time weather system to retrieve and process weather data. This data is then stored in a database for aviation decision support in general and for aircraft UPR in particular. Second, flight and weather scenarios are designed to test UPR methods and assess user preferred routes provided by these methods in this environment. Third, a segment based simulation environment is developed to simulate any type of route segment (climb, cruise, or descent). The simulation uses point mass model of aircraft, and is implemented in a continuous environment, as well as in a fast time mode. This simulation environment is applied effectively and flexibly to evaluate flight routes.

- **The impact of different emissions on the environment are effectively measured.**
  First, I developed a real time flight data management system with algorithms to process, estimate, and integrate flight data to construct the flight objects in real time, using statistic techniques and heuristics. Then another real time system is develop to estimate aviation emission using the flight objects provided by the flight data management system. The aviation emission is stored in 4-D database to analyse the impact of aviation emission on the environment. The fuel flow and emissions of a flight are calculated every time step using the speed, altitude, and phase of the flight, along with aircraft-specific aerodynamic models. These results are combined with aircraft position and time to generate a 4D (latitude, longitude, altitude and time) emission database. With these characteristics, the system provides more accurate fuel flow and emissions computation than previously possible. The real time emission data can be used to analyse the impact on the environment and support timely decision making. A number of models for emissions analysis are designed and implemented. The models developed here are then used for the calculation and analysis of fuel and emissions for UPR routes.

- **The advantages and disadvantages of the two optimisation methods (black and white box approaches) using evolutionary algorithms are**

pointed out.

Two approaches using evolutionary algorithms are designed and developed to find user preferred routes. One is the black box approach (which uses Genetic Algorithms) and the other is the white box approach (which uses Learning Classifier Systems). The two approaches are compared in the thesis.

GA is designed with an innovative and efficient chromosome representation, where a chromosome is a series of commands. These commands tells an aircraft to climb, or descend an additional altitude, or cruise an additional distance from current point. This representation makes sure the aircraft can perform and goes towards the destination. The algorithm to generate the 4-D trajectory by simulating the execution of the commands is developed. The 3-D route is then obtained by removing the time dimension from the trajectory.

An efficient design of a Learning Classifier System for multi-flight navigation is presented. A classifier is represented by a set of ordered rules, which are used to simultaneously navigate all the flights in the airspace. Navigation of a flight is based on the relation of the flight with factors of the air traffic environment such as wind, and bad weather. This system continually learns and refines the rules of classifiers by Genetic Algorithm, to discover the classifiers that navigate flights with minimal time of flying, and minimal discomfort (which is defined by bad weather level and the time duration of flights passing through bad weather area).

We found that both approaches are able to find user preferred routes for flights in an acceptable time. These approaches have the same running times. In the black box approach, user preferred routes are found with better objectives of time and discomfort, while in the white box approach the classifier outputs its rules in a symbolic representation, making the overall process transparent to the user and reusable.

- **User preferred routes are optimised efficiently for different mixes of users.**

In this thesis we propose innovative network models and algorithms to generate a dynamic network of 3-D control points using real aircraft performance and real wind data. A network is generated for a combination of an origin, a destination, an aircraft type, flight route or flight cruising altitude, and a weather environment. The term "dynamic" means when the combination changes the network changes. The 3-D network can be also considered as a 4-D network as the weight of time travelled is determined in links of the network. The advantage of the generated network is that it does not depend on departure time and user preference. Therefore it can be used efficiently to find UPR routes for flights with different departure times and user preferences.

- **The impacts of user preferred routes on environment and the deviation between the adjusted departure times and the original ones are studied.**

  The models, which are developed for fuel and emissions calculation and analysis, are applied to assess the impact of user preferred routes on the environment. In addition, given the highest purpose to optimise the time and discomfort for a flight, departure time adjustment is applied to resolve conflicts between UPR flights with each other and with non-UPR flights. The deviation of departure times from the original ones is then analysed to see the impact on flight departure time deviation when UPR routes are applied.

Chapter 2

Background

2.1 The Air Traffic Environment

2.1.1 Historical View

The air traffic management system in the past was mainly based on fixed route network and centralized. In early day of aviation, there were few planes in the airspace. They flied at low altitude and navigated from one landmark to another. They could not travel during night time. The first airway was constructed as a line of bonfires on the ground. They were then replaced with high-intensity rotating light beacons located along airways. This allowed aircraft to travel at night. However they could not work in low visibility conditions (e.g cloudy conditions), or when they were not turned on, or when they were distracted by other lights, especially in urban areas. These weaknesses led to the development of radio beacons. These beacons were located approximately 200 miles from each other to construct electronic airways. These beacons could help pilots follow to their destinations through the transmission of directional beams, allowing for navigation in limited visibility conditions. Then radar were used for navigation. The air traffic control system remained using fixed air corridors and was primarily manual. The system comprised of rules and procedures, and was under centralized control from the ground by air traffic controllers. The system was later increasingly supported by computer and the satellite system.

This structured and centralized system combing with the growth of aviation demand resulted in a number of problems regarding the efficiency, safety, delays, environmental impact, and air traffic control workload. These problems will be presented in details in the next section.

2.1.2 Present Picture

Air traffic continues growing worldwide. This leads to a substantial increase in delays and congestion. Nowadays air traffic planners face a problem that how to increase airspace capacity and flight efficiency while still ensuring the current set of safety standards [26].

The current air traffic management system with fixed route structure, where a number of routes are very busy, and air traffic growth leads to a significant delay. For example in Europe the air traffic in 2006 grew by 4.3% on 2005 and the average delay per movement, for all causes of delay, was 12.4 minutes for departure traffic; an increase of 9.5% on previous year and 12.3 minutes for arrival traffic; an increase of 11.4% on 2005 [57]. The Eurocontrol Performance Review Report for 2006 states that air traffic delays are expected to increase further, as capacity plans do not match traffic growth. There have been annually at least 15 percent of flights delaying at least 15 minutes since 2000 in United States [186].

The global accident rate was almost unchanged in the last 10 years. However, this current accident rate, applied to the industry growth, may cause one major accident per week [187]. Combining with the increased potential of conflicts at crossing points in the fixed route network, the safety may be getting higher concern. This, in turn, will lessen public confidence in the system and Civil Aviation may face a greater financial risk.

The current air traffic control system is centralized, where the separation is mainly the responsibility of air traffic controllers. This leads to capacity of the en route and transition (arrival/departure) airspace of the system is principally limited by the controller workload related to monitoring and controlling aircraft separation. As indicated in [134], the lack of airspace is not the reason of most of the traffic flow management problems in the US, but an inability of the controller to make separation predictions accurately and confidently. Hence, the key to achieving a large increase in the capacity of this airspace is to reduce controller workload [55].

Besides that the aviation industry is responsible for from 2.5% to 3% of CO_2 emissions [160]. Moreover it is more concerned that aviation emissions happen in high altitudes. The growth of aviation will make this number even higher in the future. The current air traffic management system is mainly based on a fixed route network. This limits flights to choose their best routes. A more flexible choice of flight routes in a "unfixed" route network may help to provide more efficient routes regarding minimization of time, fuel, and emissions, or maximization of passenger comfort.

All these challenges request an evolution in air traffic for sustainable development.

2.1.3 Future Concepts

In response to the ATM challenges, major air navigation service providers such as Eurocontrol and FAA indicated that the 4 dimension trajectory management (TM) will be based on User Preferred Routes (UPRs), the route with the best outcome from airspace user's perspective [58, 63]. The outcome may be with respect to the minimum time for the flight, the minimum cost, the minimum passenger discomfort, etc. The routing will be implemented in Free Route airspace while a fixed Route Network is still maintained. Here, Free Route Airspace [58] is a specific airspace where users shall freely plan their routes from an entry point to an exit point without reference to the Air Traffic Services (ATS) route network. In this airspace, flights will still conform to air traffic control. Aircraft outside of terminal areas would generally be free to fly user-preferred routes, and modify their trajectories enroute, without or with minimal intervention by air traffic control (ATC) [134, 78]. The air traffic management system will be increasingly decentralized and users can obtain more efficient routes.

The routing in free route airspace can provide the most efficient route possible in terms of minimum time, fuel, etc. This cannot be achieved in a fixed route network, where aircraft follow predefined routes and waypoints in the network. The "best overall outcome" trajectory (which is called Business Trajectory) can be obtained through flying aircraft as closely as possible to the User Preferred Route, while taking into account ATC constraints.

The workload of air traffic controller reduces as the result of providing aircraft the rights to fly user preferred route and modify the trajectories enroute. These finally can provide greater capacity, better efficiency, better predictability, and ground automation, improve safety, reduce fuel burn and emissions, and increase flexibility.

Here trajectory management is the process for establishment, agreement, update, and revision of the Business Trajectory through Collaborative Decision Making (CDM) processes between the aircraft operator, ATM service providers and Airports, except in time-critical situations, when only Flight Crew and Controller are involved [54]. It should be noted that trajectory management is a fundamental principle of Trajectory Based Operations (TBO) and is at the core of both SESAR and NextGen [100, 164, 65]. TBO uses 4D trajectories as the basis for planning and executing all flight operations supported by the air navigation service provider.

With the implementation of 4-D trajectory management based on User Preferred Routes as one of important future concepts in air traffic management, SESAR expects to achieve "*a future European ATM System for 2020 and beyond, which can, relative to today's performance: Enable a 3-fold increase in capacity which will also reduce delays, both on the ground and in the air; Improve the safety performance by a factor of 10; Enable a 10% reduction in the effects flights have on the environment; Provide ATM services at a cost to the airspace users, which is at least 50% less*" [165]. NextGen estimates that by 2018, "*NextGen will reduce total delays (in* *flight and on the ground) by about 35 percent compared with what would happen if nothing is done. That delay reduction will provide, through 2018, 23 billion dollars in cumulative benefits to aircraft operators, the traveling public and the FAA. In the process, NextGen will save about 1.4 billion gallons of aviation fuel during this period, reducing carbon dioxide emissions by 14 million tons*" [63].

2.2 Multi-objective Optimisation

In this thesis, we resolve aircraft User Preferred Route Problem. The problem deals with multiple objectives that are given weights presenting user preferences. These objectives usually conflict with each other. For example minimizing passenger discomfort may lead to the increase of time traveled. So UPR Problem can be seen as Multiple Objective Optimisation Problem and can be categorized in the broader area of aircraft trajectory/route optimisation. In this section, we review Multi-Objective Optimisation in general and then introduce techniques to elicit user preference. Next optimisation methods dealing with discrete and continuous search space and their application in aircraft trajectory/route optimisation problem are reviewed.

2.2.1 Basic concepts of Multi-Objective Optimisation

When we resolve a real world problem, we usually have to deal with multiple objectives and objectives are in conflict with each other. For example, when purchasing a car we usually have to find a way that minimizes the cost, while maximizing the quality of the car. This type of the problems is called multi-objective optimisation problems (MOPs). The task of solving MOPs is called multi-objective optimisation.

Mathematically, a general MOP is defined as follows:

$$\begin{aligned} &min && q = f(x) = (f_1(x), ..., f_k(x))^T \\ &s.t. && x \in X \end{aligned} \tag{2.1}$$

where $x \in R^n$ is a vector of decision variables, f_i, $i = 1, 2, ..., k$ are objective functions, and X is the set of $feasible\, solutions$ (called the feasible search region). The set of feasible solutions in the criterion space R^k is called the feasible region and denoted by Q:

$$Q = \{f(x)|x \in X\} \tag{2.2}$$

This problem almost always doesn't give an unique solution, i.e, an optimal solution that is the best on all objectives, but rather a set of solutions called the Pareto optimal set. A vector of decision variables (a solution) belonging to the Pareto optimal set is defined as follows.

Definition. A vector of decision variables $x^* \in X$ is Pareto optimal if and only if there does not exist another $x \in X$ such that $f_i(x) \leq f_i(x^*)$ for all $i \in K = 1, 2, ..., k$ and $f_i(x) < f_i(x^*)$ for at least one $i \in K$.

The vectors x^* (corresponding to the solutions included in the Pareto optimal set) are called nondominated. The plot of the objective functions whose nondominated vectors are in the Pareto optimal set is called the Pareto front.

However, users usually only need one solution from the set of optimal trade-off solution in practice. Therefore, the combination of searching and decision making is needed to solve MOPs [84]. Depending on the combination of searching and decision making, solving MOPs can be categorized into four main approaches [121]. The first one is no-preference approach that does not use preference information. A solution is given directly to the decision maker after solving a problem. The second one is decision making after search, or posterior approach. This approach finds the non-dominated set and then the most suitable one is determined, using the user preference. The third one is decision making before search, or priori. In this approach, the use of preference is incorporated before the optimisation process. As the result only one solution is provided at the end. The forth is decision making during search, or interactive approach. It is a hybrid approach of the second and third ones. The search is guided by a human decision making that is periodically used to refine the obtained trade-off solutions.

In this thesis, the third approach is used to find the most suitable solution for the MOP, where the importance weights presenting user preferences on objectives are predefined.

2.2.2 User Preference Elicitation

As mentioned above in order to choose the most suitable solution from a set of solutions, the Decision Maker (DM) needs to base on user preference. There are a variety of techniques to elicit user preferences. They are divided into two categories. One includes techniques to construct value functions, where decisions involves no element of risk and uncertainty. The other includes techniques to construct utility functions, where decisions involve risk and uncertainty.

In order to elicit value/utility function for solutions with multiple criteria, a value/utility function for each criteria is elicited first by a number of methods such as *bisection* and *probability-equivalence* (the most widely applied for value and utility function elicitation respectively). Then a value/utility function is elicited for all criteria, based on the importance weights of criteria, where the importance weights can be achieved by *swing weights* for example. Please refer to [72] for more information.

The formation of multiple criteria value function is different from that of multiple criteria utility function. For example the value function for two criteria is

$$v = w_1 v_1 + w_2 v_2 \tag{2.3}$$

while the utility function is

$$u = w_1 u_1 + w_2 u_2 + (1 - w_1 - w_2) * u_1 * u_2 \tag{2.4}$$

where v is the multiple criteria value function; w_1 and w_2 are the importance weights of criteria 1 and 2; v_1 and v_2 are the value functions for criteria 1 and 2; u_1 and u_2 are the utility functions for criteria 1 and 2.

If original values of criteria can present DM preference, the value function is linear. Otherwise the value function is non-linear.

2.2.3 Discrete Optimisation

The aircraft trajectory/route optimisation problem in general and UPR problem in particular can be solved in discrete search space. A graph can be predefined or artificially generated, then a shortest path algorithm is used to find the optimal route. In this section, basic concepts of graphs are presented first, followed by popular shortest path algorithm. Next, study of aircraft trajectory/ route optimisation in discrete environment is reviewed.

2.2.3.1 Basic Concepts of Graphs

It is well known that Euler, with his solution for the problem of Seven Bridges of Königsberg, established the theory of graphs in 1735. However, its first application to a problem in physical science did not arise till 1847, when Kirchhoff developed the theory of trees for its application in the study of electrical networks. Then graph theory and application become increasingly popular in various aspects of science and engineering until today.

A graph $G = (V, E)$ [180] contains two sets. The first set is a finite set V or elements called *nodes*. The second set is a finite set E of elements called *links*. Each link is identified with a pair of nodes. If the links of a graph G are identified with ordered pair of nodes, then G is called a *directed* or an *oriented* graph. Otherwise G is called an *undirected* or a *nonoriented* graph.

The symbols $v_1, v_2, v_3, ...$ are used to represent the nodes, and the symbols $e_1, e_2, e_3...$ are used to represent the links of a graph. The nodes v_i and v_j associated with an link e_l are called the *end nodes* of e_l. The link e_l is then denoted as $e_l = (v_i, v_j)$. Note that while the elements of E are distinct, more than one link in E may have the same pair of end nodes. All links having the same pair of end nodes are called *parallel links*. Further, the end nodes of an link need not be distinct. If

$e_l = (v_i, v_i)$, the the link e_l is called a $self - loop$ at node v_i. A graph is called a *simple graph* if it has no parallel links or self-loops. A graph G is of *order* n if its node set has n elements [180].

A *walk* in a graph $G = (V, E)$ is a finite alternating sequence of nodes and links $v_0, e_1, v_1, e_2, ..., v_{k-1}, e_k, v_k$ beginning and ending with nodes such that v_{i-1} and v_i are the end nodes of the link e_i, $1 \leq i \leq k$. Alternately, a walk can be considered as a finite sequence of nodes $v_0, v_1, v_2, ..., v_k$, such that $(v_{i-1}, v_i$, $1 \leq i \leq k$, is an link in the graph G. This walk is usually called a $v_0 - v_k$ walk with v_0 and v_k referred to as the end or terminal nodes of this walk. all other nodes are internal nodes of this walk. Note that in a walk, links and nodes can appear more than once [180].

A walk is open if its end nodes are distinct; otherwise it is closed.

A walk is a *trail* if all its links are distinct. A trail is open if its end nodes are distinct; otherwise, it is closed.

An open trail is a *path* if all its nodes are distinct.

2.2.3.2 Shortest Path Algorithms

Let G be a connected directed graph in which each directed link is associated with a real positive number called the *length* of the link. The length of an link directed from a node i to a node j is denoted by $w(i, j)$. If there is no link directed from node i to node j, then $w(i, j) = \infty$. The length of a directed path in G is the sum of the lengths of the links in the path. A minimum length directed $s - t$ path is called a *shortest path* from s to t. The length of a shortest directed $s - t$ path is called the *distance* from s to t, and it is denoted as $d(s, t)$. Clearly, $d(i, i) = 0$ for all i.

There are many algorithms which can be used to find the shortest paths. They may be used to find the shortest paths from a node s to another node t, or from a node s to all other nodes in G, or between all the ordered pairs of nodes in G. These algorithms are listed below.

- Depth first search: DFS traverses or searches a graph, starting at the root then exploring as far as possible along each branch before backtracking. The complexity of DFS is $O(|V| + |E|)$ time. The algorithm is complete. [38]

- Breadth first search: BRFS begins at the root node and explores all the neighbouring nodes. Then for each of the nodes, it explores their unexplored neighbour nodes,.. Until it finds the goal. The complexity of the algorithm is $O(|V| + |E|)$. The algorithm is complete. [107]

- Dijkstra: The algorithm starts with the object's starting point. Then repeatedly examines the closest not-yet-examined node, adding its nodes to the set of nodes to be examined. Expands outwards from the starting point until it reaches the goal. The complexity of the algorithm is $O(|V|^2)$. The algorithm is complete. [38]

- Bellman-Ford: The main idea of the algorithm is that the distance of any node to the starting node is equal to the minimum of all neighbouring node distances plus the length of the corresponding path. The algorithm solves this by successive approximations. Bellman-Ford is for graphs without negative circles that are reachable from the starting node. The complexity of the algorithm is $O(|V||E|)$ and the algorithm is complete. [38]

- Floyd Warshal: Assume that we have a function $SP(i, j, k-1)$ which is the shortest possible path from i to j using only nodes 1 through $k-1$ as intermediate points along the way. The shortest path from each i to each j using only nodes 1 through k is

  $$SP(i, j, k) = min(SP(i, j, k-1), SP(i, k, k-1) + SP(k, j, k-1))$$

  The algorithm works by first initializing $SP(i, j, 0)$ for all (i, j) pairs by the equation below:

  $$SP(i, j, 0) = w(i, j) \tag{2.5}$$

Then using that to find $SP(i, j, 1)$ for all (i, j) pairs, etc. This process continues until $k = n$, and we have found the shortest path for all (i, j) pairs using any intermediate nodes.

The complexity of the algorithm is $O(|V|^3)$ and the algorithm is complete. [38]

- Johnson: The algorithm makes improvement for sparse graphs with reweighting technique. If all weights are non-negative, running Dijkstra with each node as starting node else reweighing. The complexity of the algorithm is $O(|V|^2 log(|V|) + |V||E|)$ and the algorithm is complete. [38]

- Hill climbing 1: The algorithm performs depth-first, but instead of left-to-right selection, the algorithm selects the child with the best heuristic value first. The complexity of the algorithm is $O(|V| + |E|)$ and the algorithm is complete. [154]

- Beam search: Assume a pre-fixed $WIDTH$, the algorithm performs breadth-first, but only keep the $WIDTH$ best new nodes, depending on heuristic at each new level. The complexity of the algorithm is $O(WIDTH * d)$ (d is maximum depth) and the algorithm is not complete. [166]

- Hill climbing 2: The algorithm is beam search with a width of 1.

- Local search: Hill climbing 2 is an example of local search. In local search, we only keep track of 1 path and use it to compute a new path in the next step.

- Best first search [154]: The algorithm always expands the heuristically best nodes first. The Best-First-Search (BFS) algorithm works in a similar way to Dijkstra, except that

  - It has some estimate (called a heuristic) of how far from the goal any node is.

  - Instead of selecting the node closest to the starting point, it selects the node closest to the goal.

  - BFS is not guaranteed to find a shortest path

- A* Search [45]: A* is like:
  - Dijkstra's algorithm in that it can be used to find a shortest path
  - BFS in that it can use a heuristic to guide itself

  A* computes the function $f(n) = g(n)+h(n)$, where $g(n)$ is the actual shortest distance traveled from initial node to current node, and $h(n)$ is the estimated (or "heuristic") distance from current node to the goal, then $f(n)$ is used to determine the order in which the search visits nodes.

2.2.3.3 Aircraft trajectory/route optimisation in discrete search space

A graph is constructed in the airspace through the predefinition or artificial generation of nodes (or waypoints) and links between these nodes. In [41] the model for the airspace is composed of a set nodes, and the routes are formed by computing the Delaunay triangulation of these nodes. In [133] a rectangular gridding system is used to represent the airspace, where each grid point is considered as a waypoint. In [202], the airspace is divided to a square grid (2-D grid), nodes are located in the cross-points of the grid. The network is constructed by connecting the nodes to plan for military flights for threat avoidance in no-wind environment. In [128], the airspace is divided to 3-D cubes. The network is constructed by connecting nodes of cubes. The network generation is for military flights and does not consider wind. Some research using a fixed or dynamic route network is particulary to find optimal routes in the transition/terminal area around a airport [141, 5]. The wind information is also not taken in these studies. In this thesis, we present a novel and efficient network model and algorithms to generate a network, taking aircraft performance and weather information into consideration for finding optimal aircraft routes from origins to destinations.

2.2.4 Evolutionary Computation

Finding optimal solutions for a real world problem may be computationally expensive or even impossible, as the complexity of the problem presents exact methods from being applicable. Finding the aircraft optimal trajectory/route in continuous airspace is such kind of problem. Evolutionary Algorithms (EAs) are broadly used to these types of problem. These algorithms are stochastic search methods. They try to find a good approximation but do not guarantee to find the optimal solutions. In this section we will present two major sub-fields of EAs: Genetic Algorithms (GA) and Learning Classifier Systems (LCS). GA outputs the descriptive representation of solutions as an arrays of bits, integers, floats or data structures. From these arrays, solutions are usually obtained through simulation. LCS is a machine learning system, based on Genetic Algorithms. LCS consists of a population of rules. A Genetic Algorithm is used to evolve the population to find the best rules. LCS outputs rules. Solutions are then inferred from these rules. Evolutionary Multi-objective Optimisation, which is suitable for Multi-Ojective Optimisation Problem, is presented at the end of this section.

2.2.4.1 Genetic Algorithms

(a) Genetic Algorithm Concept

Genetic algorithm (GA) was first proposed by Holland in 1975 [81]. It is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimisation and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimisation problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

GA evolves a population of strings (chromosomes or the genotype of the genome), which encode candidate solutions (individuals, creatures, or phenotypes) to an optimisation problem, toward better solutions. A solution is originally represented by a binary string. Then other encodings [181] are also applied such as real number, order-based, embedded lists, variable element lists etc. A population is normally initialized randomly. Then it is evolved through generations. In each generation, the fitness of every individual in the population is calculated. A set of individuals are selected from the current population by stochastic methods, based on their fitness. A new population is generated by modifying the set of selected individuals (recombination or mutation). There are many criteria to terminate the algorithms. The two most common criteria are the maximum number of generations that the algorithm produces, or fitness level satisfactory the current generation achieves.

Figure 2.1 presents a simple generational genetic algorithm procedure.

Step 1. Initialize a population of individuals.

Step 2. Evaluate the fitness of each individual in the population.

Step 3. Select individuals for reproduction.

Step 4. Reproduce new individuals by crossover and mutation operators.

Step 5. Evaluate the fitness of new individuals.

Step 6. The new population may be the set of new individuals or the combination between the current population with the set of new individuals.

Step 7. If terminating condition is satisfied Stop, otherwise goto Step 3

Figure 2.1: Simple generational genetic algorithm procedure

Initialization

The individuals of the initial population is usually generated randomly. The number of individuals depends on the problem. Typically the population contains several hundreds to thousands of possible individuals. The randomly generated population can help to explore the entire search space. However, some research may seed solutions in promising areas, where the likelihood of optimal solutions existing in is high.

Evaluation

Fitness value is to show how good an individual is in relative comparison with other individuals in the population. Fitness function can be objective function of the problem or a different function.

In some complex problems, the exact fitness function may request a lot of time to evaluate (such as a number of hours). Furthermore the fitness evaluation is repeated in every individual of a population through generations. Typical genetic algorithms cannot resolve such types of problem. In this case, an approximated fitness that request less time to evaluate can be a solution. Besides that, some fitness function cannot be defined explicitly. Human interaction is required for assessing the goodness of solutions. This approach is known as Interactive Genetic Algorithm or Human-Based Genetic Algorithm [108], which has human interfaces for initialization, mutation, crossover, selection, and evaluation.

Selection

In each generation, a pool of individuals are selected from the current population to breed a new generation. Selection is implemented, based on the fitness of individuals. Some selection methods rank the fitness of each solution, and the best solutions are preferentially selected. However this method may consume a lot of time. Other methods only rank a number of random individuals.

Reproduction

In reproduction process, a new population is created from the pool of selected individuals by genetic operators (crossover and/or mutation). Biologically inspired, two "parent" solutions are selected from the pool to reproduce a new child until the the set of offsprings reaches a predefined size. However, some research suggests to use more than two parents to breed a good child [53, 185].

The probabilities of crossover (pc) and mutation (pm) play an important role in determining the degree of solution accuracy and the convergence speed that genetic algorithms can achieve. Generally pc is high, while pm is low (where the role of mutation is to avoid the loss of potential solutions). However, some research uses mutation as the main operator because the crossover operator may not be useful in some cases. For example, when individuals in a population are similar, the crossover operator will create children similar to their parents. Some genetic algorithms with adaptive parameters (AGAs - Adaptive Genetic Algorithms) use the population information to adjust the pc and pm to obtain the population diversity and convergence. The population information can be simply the fitness values of the solutions [174] or the optimisation states, which can be assessed through clustering analysis [204].

In addition to the two main operators: crossover and mutation, other genetic operators have been used such as regrouping, colonization-extinction, or migration [2].

The new population may be the set of new individuals. This is known as non-elitist selection. Another strategy is known as elitist selection, where the new population also includes the better solutions from the current population.

Termination

The evolution terminates until a certain condition has been satisfied. Commonly, the terminating conditions are a found solution meeting minimum criteria, or reaching maximum number of generations, or reaching allocated budget (such as computation time), or successive iterations no longer producing better results. Manual inspection and combinations of these conditions can be applied.

Genetic algorithms have been applied in many fields such as bioinformatics, phylogenetics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields. The major requirements, when applying GA in a real world problem, are the representation and evaluation of solutions. Once the genetic representation and the fitness evaluation are defined, a GA can simply follow a set of standard mutation, crossover, and selection operators to evolve a population of individuals towards better solutions. In this thesis we apply GA to aircraft User Preferred Route Problem.

(b) Aircraft Trajectory/Route Optimisation using GA

There have been many studies, which apply GA to resolve aircraft trajectory/route optimisation. The main designs in these study are the representation and simulation of aircraft routes. In [142], the flight route, which connects A and B, is supposed a single value function; Fourier transform is applied for it and keep the low frequencies parts due to the constraint that the aircraft only maneuvers at low altitude, it is found that the flight route can be described as a series of sine waves with angle frequencies and amplitudes.

In [70], authors use the differentiable quadratic spline wavelets for decomposition of an Unmanned Aerial Vehicle (UAV) route into wavelet coefficients. With a wavelet-based route representation, the trajectory optimisation problem convert to the problem to identify the wavelet coefficients. When the multiple wavelets are combined, the generated trajectory satisfies mission requirements (hitting a target, avoiding threats, not flying into terrain, avoiding severe weather) while ensuring that the maneuvers are flyable and, some mission specific criterion (e.g., minimum altitude flight, or minimum fuel expenditure) is optimised.

In [127] B-Spline curves are used to represent the UAV path.

In this thesis, we propose another innovative and efficient way to present and simulate aircraft routes.

2.2.4.2 Learning Classifier Systems

(a) Learning Classifier Systems Concept

A Learning Classifier System (LCS) is a machine learning system [126] related to reinforcement learning [179] and Genetic Algorithms. The origin of LCSs can be seen in Holland's work on complex adaptive systems and his early proposal on schemata [79]. This established the basis for the first practical implementation of a classifier system, which was called CS-1 (Cognitive System Level One) [80]. CS-1 had capability to capture and react appropriately to events from the environment. This system maintained a population of binary rules. Genetic Algorithm was used to adjust and select the best rules, where a rule fitness was evaluated using a reinforcement learning technique. The system inspired a stream of research, named as the Michigan approach. Another line of LCSs, named as the Pittsburgh approach, was also investigated simultaneously with the early developments of Michigan LCSs.

In Michigan approach, a population is a single set of rules. The GA evolves and selects the best classifiers from that set. A population of rules is generated first. A reinforcement-learning technique is used to evaluate rules through credit assignment. The bucket brigade algorithm [80] was a popular algorithm for credit assignment. This algorithm assigns the strength for rules, based on the predicted payoffs that the environment gives to them. GA searches towards the best rules (which are given high reward) by considering the strength of a rule as its fitness, while maintaining the diversity of the population by using niching techniques and a non-generational scheme [71].

The Pittsburgh approach was emerged with the LS-1 classifier system [171], which was the inspiration for the classifier scheme GABL [99]. In Pittsburgh approach, LCS has a population of rule-sets. The GA task is to recombines and reproduces these rule-sets to find the best rule-sets. The fitness evaluation of a rule-set is based on the performance of its rules.

In Pittsburgh LCS each individual is a complete rule set, which eliminates the need for cooperation among individuals required in the Michigan approach. A such, the operation of the GA is simpler, the GA does not need to converge to a single solution. However, since Pittsburgh LCSs searches in the space of feasible rule-sets, the search space is larger and usually requires more computational resources than the Michigan approach. In addition, few controls can be exercised on the rule level. Two additional operators were designed in GABIL to overcome these issues. GIL [95] was another proposal that included a large set of operators acting at different levels with the aim to achieve control over the type of rules evolved, but an increased parameterisation cost is paid.

While an increase in flexibility, resulted from the use of variable sized individuals, parsimony pressures [69], and the use of multiobjective fitness [113, 15], were necessary to control the excessive growth of individuals.

(b) Aircraft Trajectory/Route Optimisation using LCS

There have been studies on designing LCS for aircraft trajectory/route generation. Genetic-based machine learning techniques are used for multi-agent route planning in discrete environments, where the movement of one agent is from one cell to another cell in an environment of grid cells [35]. In [170], a genetics-based machine learning system is implemented to acquire rules for novel fighter combat maneuvers through simulation. The LCS is designed for discovering rules that dictate combat maneuver logic rather than presenting the interaction between flight and weather environment to achieve optimal routes for civilian flights in terms of time, fuel, emissions or comfort. This thesis designs and develops LCS for user preferred routes for civilian flights in continuous environment to achieve such objectives.

2.2.4.3 Evolutionary Multi-objective Optimisation

As presented above finding optimal solutions is hard for some real world optimisation problem. It is harder for multi-objective optimisation problem as the algorithm needs to deal with multiple objectives (usually conflicting) and the output is a set of non-dominated solutions rather than a single solution. EAs with stochastic and population based optimisation mechanism are particularly suitable to solve multiobjective optimisation problems. A set of candidates are maintained in these algorithms. Therefore several solutions of the Pareto set are found in a single run of the algorithm, while traditional mathematical programming techniques request a number of runs. EAs are also less affected by the shape or continuity of the Pareto front. It is easy for EAs to deal with discontinuous or concave Pareto fronts, while it is not for mathematical programming techniques.

Multi-Objective Evolutionary Algorithms (MOEAs) have to design to guide the search towards the Pareto Set and to keep the diversity of non-dominated set [42]. These two goals are conflicting. while the first goal is minimizing the distance from the generated solutions to the Pareto set, the second is maximizing the diversity of the obtained Pareto set. It is also not easy to define the closeness to the Pareto set and the diversity of the Pareto set.

The first goal is mainly related to the selection of individuals for reproduction, in particular to the problem of assigning scalar fitness values in the presence of multiple optimisation criteria. The second goal deals with selection in general to avoid the identicalness in the population (most of individuals are identical) regarding to both objective space and decision space. Finally, a third issue addressing both of the goals is elitism to prevent nondominated solutions from being lost.

Fitness Assignment

While fitness function is the same to objective function for single-objective optimisation, fitness function and selection involve in several objectives for multi-objective optimisation problems. There are generally three ways to assign fitness: aggregation-based, criterion-based, and Pareto-based [205]. Aggregation-based fitness assignment approaches are to aggregate the objectives into a single parameterized objective function. The parameters of this function are systematically altered during the optimisation process to find a set of nondominated solutions instead of a single trade-off solution. A popular way is weighted-sum aggregation, where the weights represent the changing parameters during the evolution process [93]. In criterion-based approaches, each time an individual is selected for reproduction, potentially based on a different objective. The objective may be chosen randomly, or based on its distinct level, or a predefined probability [109, 159]. Pareto-based approaches use dominance relation among individuals to assign fitness values. In these approaches, fitness assignment is related to the whole population, while aggregation based approaches concern only the individual itself. In Pareto-based approaches, the fitness assignment can be based on dominance rank (the number of individuals an individual is dominated) [68], dominance depth (the population is divided into several fronts and the depth reflects to which front an individual belongs to) [175, 43], dominance count (the number of individuals dominated by a certain individual), or a combination of these dominance types [207].

Diversity Preservation

The diversity of the current pareto set is maintained through selection process. The probability of an individual being selected is lower, the higher the density of its neighbor individuals. MOEAs use techniques in statistics to estimate density such as Kernel methods [68], Nearest neighbor techniques [206], and Histograms [105]. In Kernel methods [169], the density around an individual is estimated by the sum of the values of Kernel function, which has the argument being one of the distances from the individual to all other individuals. In nearest neighbor techniques [169], the density is estimated, based on the distance from an individual to kth nearest neighbor individuals [169]. A function of the inverse of this distance is usually used for the estimation. Histograms [169] divide the space by a hypergrid. The density around an individual is estimated by the number of indviduals in the same box of the grid.

Elitism

Elitism is to keep the best individuals through generations. This can be obtained by combining the current population and the offspring to create a new population, where the promising individuals from both current population and offspring are preserved. Another way is maintaining an archive of promising solutions. Archive members can be used in the selection process. However, the archive may just work as an external storage (may be not taken into account during selection). The selection of individuals to be preserved is usually based on dominance and density information. The non-dominated solutions are preferred to the dominated ones, and the less density solutions are preferred to the high density ones. Algorithms such as PAES [106], SPEA2 [206], PDE [1], NSGA-II [44], and MOPSO [37] are typical in this category.

In contrast, the non-elitism does not clearly preserve good solutions from the current population for the next generation [42]. The population of the next generation is created by all crossover and mutation operators from the mating pool, which contain individuals selected from the current population. In [36], all algorithms using this approach are illustrations of the first generation of MOEAs. The only difference from conventional EAs is that they use the dominance relation, when solutions are assessed. Illustrations of this category include MOGA [68], NPGA [85] and NSGA [42].

In this thesis, we use weighted sum technique to convert the multiple objective optimisation problem into single objective optimisation problem. Therefore we don't use any EMO techniques to resolve UPR problem in this thesis. However, these EMO techniques will be considered to use in the future to deal with objectives that we cannot simply convert them to single objective optimisation problem by weighted sum technique.

2.3 Aircraft Trajectory Management

Trajectory Management (TM) in general and User Preferred Routing (which is the base for TM) in particular need to use information such as airspace, flight, weather, and terrain data. These information is determined, and broadcasted, based on air traffic control technology (e.g communication, navigation, and surveillance systems). In this section, we mainly present technology that will be used to support the future operational concept. UPR Planning can be categorized in broader areas that are trajectory or route optimisation with the same objectives such as minimizing flight cost, guaranteeing flight safety or avoiding conflicts. For safety management, the potential aircraft conflicts can be generally minimized through Air Traffic Flow Management (ATFM). In the following sub-sections, we will present these closely related areas of TM as well as UPR planning in details.

2.3.1 Technology enablers and airspace data

While the routing will be implemented in Free Route airspace, a fixed Route Network is still maintained. In this section, while presenting technology enablers mainly for the future TM, we present airspace data belonging to the traditional fixed route network.

2.3.1.1 Technology

In order to manage air traffic in general and to generate aircraft routes in particular efficiently, relative information needs to be provided timely, accurately, and to the right people. The more updated accurate information the better planning. These information can be flight, weather, or terrain information. Technology enablers (such as communication systems, navigation systems, and surveillance systems, which meet the identified operational and architecture requirements in providing and distributing the information in time and to the right location with the required availability, continuity and integrity) have been identified by FAA and Eurocontrol [63, 165].

The communication systems will increasingly use digital/data technology and protocols. A full integration of terrestrial and satellite networks towards System Wide Information Management (SWIM), connecting all ATM sub-systems will be established. SWIM will carry digital information. It provides the IT infrastructure necessary for air traffic systems to share information, increase interoperability, and encourage reusability of information and services. SWIM will decrease the reliance on voice communication and significantly reduce opportunities for error. It will enable cost-effective, real-time data exchange and sharing among users. SWIM will improve agility to deliver the right information to the right people at the right time.

The primary navigation system will be based on satellite. When satellite navigation services are not available, a fall back solution is provided.

The surveillance systems will be based on Automatic Dependent Surveillance Broadcast (ADS-B). They will increasingly provide improved 4D-position information (3-D spacial position and time). Utilizing Global Positioning System (GPS) satellite signals, ADS-B helps pilots and air traffic controllers to locate and control aircraft more precisely and in real time. The position information combines with other data and broadcast out to other aircraft and air traffic control facilities.

Through ADS-B, pilots can access to weather services, terrain maps and flight information services. This helps them improve situational awareness and maintain a safe separation from one another with less assistance from controllers.

2.3.1.2 Airspace data

The traditional airspace structure is still available with airspace data such as sector, waypoint, airway, and route.

A sector is a piece of airspace with a polygon boundary as lateral limits and with defined bottom and top altitudes/levels as vertical limits. The airspace controlled by a controller is his/her sector [39]. Sector capacity is one important factor in aircraft route planning as indicated in [41].

A waypoint is a reference location, which is used for navigation. Waypoints may correspond to a specific latitude/longitude or may be conditional (also called floating). Waypoints can be specified in a variety of manners such as: latitude/longitude, fix-radial-distance, intersection of two radials, etc [64]. A conditional waypoint is conditional on the trajectory (e.g., turn to heading 270 upon crossing 2000 feet). In another definition, a way point is a predetermined geographical position used for route definition and/or progress reporting purposes that is defined by latitude/longitude [60]. A fly-by waypoint requires the use of turn anticipation to avoid overshoot of the next flight segment. A fly-over waypoint precludes any turn until the waypoint is overflown and is followed by an intercept maneuver of the next flight segment.

Airway is a control area or portion thereof established in the form of a corridor equipped with radio navigation aids [56]. The earliest airways in the United States were constructed by the US Post Office to guide airmail pilots on their delivery routes. These airways were between major cities and identified at night by a series of flashing lights and beacons, which pilots flew over in sequence to get from one city to the next.

In a joined document between FAA and Eurocontrol in 2004 [64], a route is a combined series of way points and airways, that specifies the approximate lateral path to be followed by a flight. The way points are approximate since turn dynamics will result in aircraft flying by, or over, a way point. It is also known as the host track defined by the way points in the active flight plan. However, more recently in 2007, in Concept of Operations for the Next Generation Air Transportation System [100], a route is a 3D path through space without time component. In SESAR Definition Phase: The Concept of Operations at a glance [164], a route can either be a 2D route (with lateral containment), or 3D routes (with lateral and vertical containment). This means a route can specify both lateral and vertical profiles.

2.3.2 Flow Planning Tools

As presented in Section 2.1.3, air traffic congestion is a critical problem. The costs impact on both airlines and passengers is huge. Congestion happens as the consequence of the exceed of airport and sector capacities. It usually occurs in peak travel times in a year and in a day, or in bad weather period. The long-term solutions can be constructing new airports and runways, using larger aircraft, using advances in ATC. In the short-term, air traffic flow management (ATFM) is the best available way for delay reduction [22]. ATFM is the regulation of air traffic in order to avoid exceeding airport or air traffic control capacity in handling traffic, and to ensure that available capacity is used efficiently. ATFM can be classified into two categories: strategic and tactical TFM.

Strategic TFM is taken before aircraft take off through route planning, and ground holding. The routes of scheduled flights can be strategically planned in advance to ensures that the distribution of traffic flows can minimize congestions. Ground holding is the action of delaying take-off. Ground holding helps to translate predicted air-borne delays to the ground. The reason for doing so is the costs of airborne delays much higher than ground delays. An aircraft in the ground consumes less fuel, and request less control effort (be safer) than in the air. It can be considered as a way to reduce controller workload by limiting the number of aircraft, which are airborne at any given time. There have been massive studies on Ground-Holding Problem (GHP). Models in GHP attempt to assign ground holding delays to flights, with the objective of minimizing the cost of delays, while satisfying any existing constraints on ATM capacities at airports or en route. The GHP can be categorized into two subproblems, the Single-Airport Ground-Holding Problem (SAGHP) and the Multi-Airport Ground-Holding Problem (MAGHP). In the SAGHP, ground holding times are assigned to flights scheduled to travel to some particular airport, where scheduled demand is expected to exceed available capacity during some period of time. In the MAGHP, delays are assumed to propagate in the network of airports, as aircraft perform consecutive flights, thus the examination of an entire set of airports simultaneously is necessary [22]. The GHP can be further categorized into a "deterministic" version (deterministic GHP) and a probabilistic version (stochastic GHP). The deterministic GHP is resolved with assumption that the sector and airport capacities and weather can be predicted with perfect accuracy. The stochastic version arises as the GHP is often solved in the considerable uncertainty of sector and airport capacities, and weather. Small changes of weather may cause large differences in sector and airport capacities. So far meteorologists are almost impossible to forecast weather changes with high accuracy, even in a short time-period of an hour or less. Therefore, ground-holding decision making is under uncertainty. When a decision is made, there needs to consider the trade-off between "conservative" strategies which may have assigned times exceed ground-holds and more "liberal" ones, which may have expensive airborne delays [22]. There are three models to resolve the SAGHP: deterministic model [182, 183], stochastic model [11] and both deterministic and stochastic model [153, 152]. For the MAGHP, all models are deterministic ones [22] such as in [184, 193, 10, 23].

Tactical TFM is taken while aircraft are airborne though re-routing aircraft in "real time", possibly changing flight plan, speed controlling, and airborne holding en route and, especially, in the vicinity of destination [184, 193, 23].

Various algorithms have been proposed for the above TFM models and actions such as dynamic programming [182], integer programming [193, 23], stochastic programming [151], and heuristic methods [182]. In this thesis, we use taking-off time adjustment as a flow planning technique to resolve conflicts among flights.

2.3.3 Trajectory and Route Optimisation

2.3.3.1 Trajectory and Route Smoothing Techniques

In order to find the best route meeting user preference, a route needs to be simulated for evaluation as well as to be smooth to satisfy the curvature and speed constraints.

In [8, 34, 49, 199], the shortest path between two points are constructed as combination of circles and straight-line path segments to meet curvature constraints.

In [127], B-Spline curves are used to smooth flight route in 3-D space. A number of similar techniques can be also considered to smooth aircraft trajectory such as Lagrange Interpolating Polynomial [176], Least Squares Fitting [66], Least Squares FittingExponential [66], Least Squares FittingPolynomial [66], Nonlinear Least Squares Fitting [17], Polynomial Curve, Spline [16], and Bzier Curve [16].

In [142], a flight route is described by a series of sine waves with different frequencies and amplitudes.

Neural networks can be applied to the problem of curve fitting [25]. In order to smooth aircraft trajectory, neural networks are first trained by a set of recorded trajectories. Then these neural networks are used to generate unavailable points of a trajectory, which is given a reduced set of known points. The set of generated points will be best fit the set of know points. In [167], neural network has been applied to predict the future aircraft altitude and speed, based on the set of last points.

Kalman filter is a mathematical method to use the measured values containing noise and other inaccuracies, and produce values that tend to be closer to the true values of the measurement. In [12], Kalman Filters has been used for tracking a maneuvering aircraft, this technique can be further developed to smooth aircraft trajectory.

In this thesis, we have not used any above techniques to smooth a flight route for curvature constraint salification. However, for speed constraint aircraft speed is generated based on its altitude and BADA (The Base of Aircraft Data) database for evaluation of aircraft route. In our future work, we will consider to apply the above techniques to smooth aircraft routes.

2.3.3.2 Trajectory and Route Optimisation Techniques

In this section, we review the problem of aircraft trajectory/route optimisation (ATRO) problems and the methodologies for resolving these problems in literature. In route optimisation problem, a state of aircraft is defined by coordinates (latitude, longitude, and/or altitude) (see aircraft route definition in Section 2.3.1.2), while in trajectory optimisation problem a state of aircraft includes not only coordinates but also time (see the definition and attributes of 4D trajectory in [100]) or in other words in addition to optimising spatial trajectory, speed, turning rate and turning angle of aircraft also need to be optimised.

The UPR problem is a branch of aircraft route optimisation, where aircraft route optimisation deals with multiple objectives and objectives are given weights in this thesis. From the literature, we found that there have been a lot of study in this field dealing with single objective. However there are few dealing with multiple objectives. As pointed out by Mittal and Deb [127] in 2007, all the existing methods for path planning do not take into account the fact that the problem involves a number of conflicting objectives which must be taken into consideration while generating a path, especially for aircraft path planning problem which is formulated to deal with multiple aircraft in a weather environment.

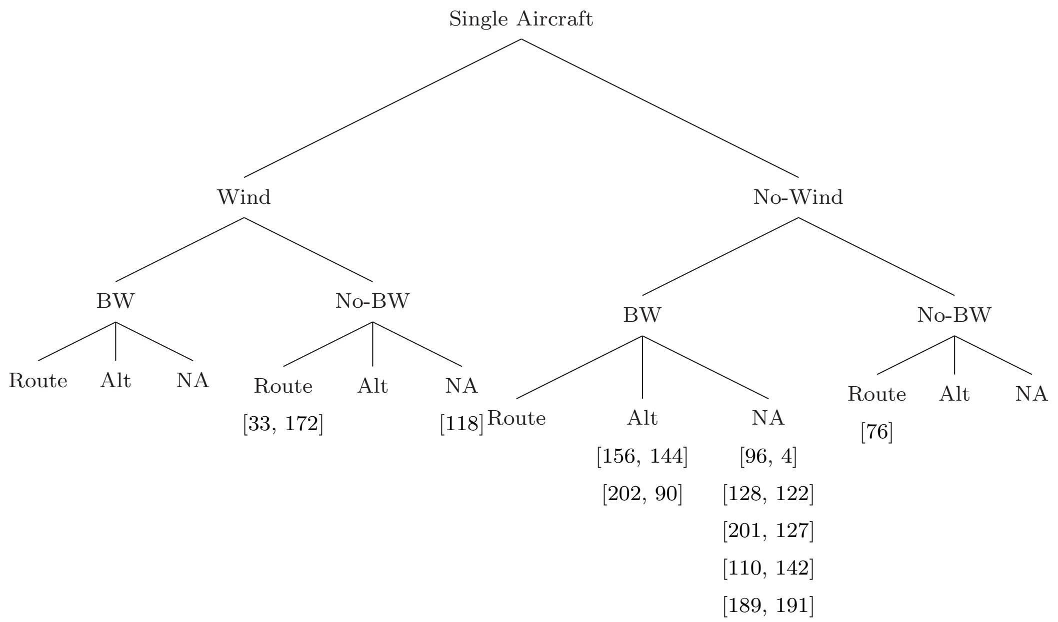
(a) Category by Problem Definitions

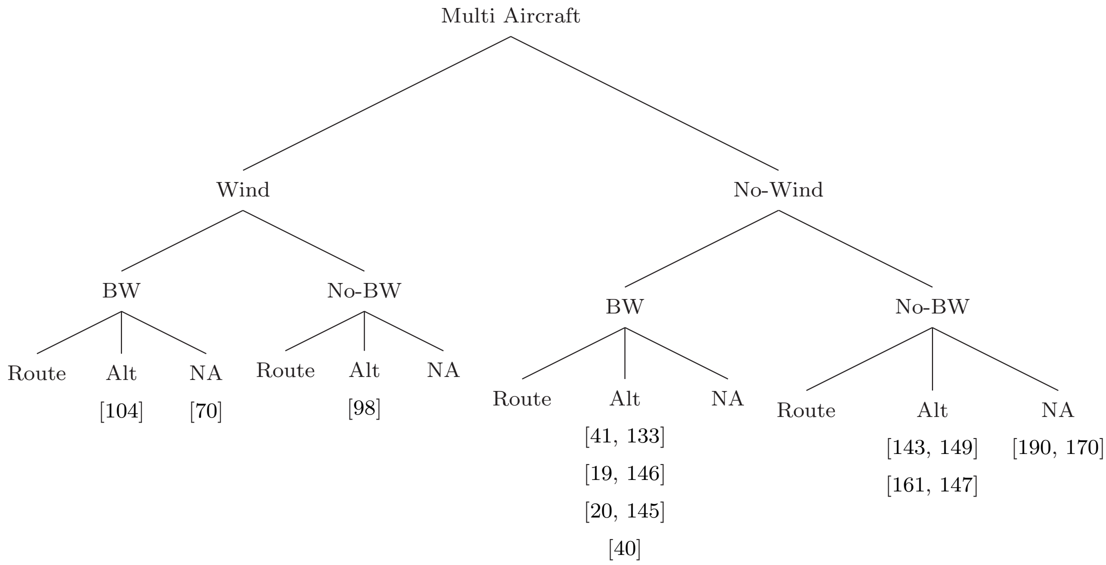
ATRO problems are categorized by input data into 2 different trees for Single Aircraft, and Multiple Aircraft. Each tree continues branching by other input data:

- Whether the problem considers wind.
- Whether the problem considers bad weather, threat, or obstacle.
- Whether a flight is given cruising altitude, route or not given these information.

In these trees, "BW" can be considered as bad weather, threat, or obstacle flights need to avoid. "Alt" means the cruising altitude of a flight is given. "Route" means the route of a flight is given. "NA" means both route and cruising altitude are not given.

The current ATRO problem category trees show that there is a number of study on the problem considering wind and storm. However, most studies deal with single objective only and user preferences are not taken into account. For example, only time is minimized in [104]. In [70] only obstacle avoidance is considered.





In these above trees, study in [202, 4, 201, 127, 110, 142, 191, 70, 98, 41, 133, 170] is to optimise aircraft route, while study in [33, 172, 118, 156, 144, 90, 96, 128, 122, 189, 76, 104, 19, 146, 20, 145, 40, 143, 149, 161, 147, 190] is to optimise aircraft trajectory.

(b) Category by Algorithms

Many methods have been proposed to resolve aircraft route and trajectory optimisation problems. These methods are based on a number of approaches which are categorized as follows.

Mixed integer linear programming Aircraft trajectory optimisation including collision avoidance can be written as a linear program subject to mixed integer constraints known as a mixed-integer linear program (MILP). This can be solved using commercial software written for the operations research community [149, 19, 161, 156, 145, 110, 40, 104, 147]. In these studies aircraft is approximated as a point mass, a state of aircraft may include geographic position, velocity, turning rate and turning angle. Approaches based on MILP for trajectory optimisation are in [149, 161, 156, 145, 40, 147] and those for route optimisation are in [110].

Nonlinear programming Nonlinear programming (NLP) is the process of solving a system of equalities and inequalities, collectively termed constraints, over a set of unknown real variables, along with an objective function to be maximized or minimized, where some of the constraints or the objective function are nonlinear [21]. Aircraft trajectory optimisation problem is formulated as optimal control problem and then it can be resolved by NLP. Approaches based on NLP for trajectory optimisation are in [190, 96, 192, 156, 90, 191, 150].

Singular perturbation In [144], methods of time-controlled optimal flight trajectory generation that included the effects of risk in a threat environment are developed. Singular perturbation (SP) theory [83] is used to obtain a reduced order airplane model of the transport class and then Pontryagin's Minimum Principle [140] is used with a Fibonacci search method [67] to minimize a cost functional that consists of a weighted linear combination of risk, fuel-flow, time, and position constraints. The paper presents research results on a method to perform waypoint rendezvous in conjunction with multiple-threat avoidance with the horizontal plane trajectory optimisation algorithm. The approach based on singular perturbation for trajectory optimisation can also be found in [189].

Petal Algorithm The petal algorithm uses a heuristic method to prune out the solutions that are not likely to be part of the optimal solution, which significantly speeds up the task assignment process. This technique is used to find optimal aircraft trajectory in [146].

Graph Based Algorithms The optimal trajectory or route of a flight is searched in a graph of nodes which may be real waypoints or artificial waypoints and arcs connecting nodes. The search algorithms can be Dynamic Programming, Dijkstra, Label Setting Algorithm, etc. These approaches for trajectory optimisation can be found in [128, 20] and those for route optimisation can be found in [41, 133, 202, 201, 111].

Sequential quadratic programming In [122], a approach based on a combination of nonlinear control theory, spline theory, and sequential quadratic programming is proposed to generate trajectories in real-time.

Evolutionary Algorithms Many methods based on Evolutionary Algorithms to find optimal routes for aircraft. For example, in [142, 70, 127] methods based on Genetic Algorithms are used; a method based on Ant Colony Optimisation is used in [4]; and in [170] a method based on Classifier Systems is proposed.

Analytical optimisation approach based on calculus of variations In [202, 201], analytical optimisation approach for routing an aircraft in a threat environment have been developed. Using this approach, an aircraft's optimal risk trajectory with a constraint on the path length can be efficiently calculated. The analytical approach based on calculus of variations reduces the original risk optimisation problem to the system of nonlinear differential equations.

Optimal Control In [98] Neighboring Optimal Control(NOC) is applied to the problem of optimising aircraft route in a general wind field. The family of sequential gradient-restoration algorithms for optimal control problems is used for Take-off optimal trajectory problem in wind shear in [119]. Dual sequential gradient-restoration algorithm (DSGRA) for optimal control problems is used for landing optimal trajectory problem in wind shear [120].

Decision analysis In [191], an approach for modeling and solving an aircraft trajectory optimisation problem by the methods of decision analysis [196, 102, 198] is introduced.

(c) Category by Phases

There are a variety of study on ATRO. Some optimise the whole trajectory/route from origin to destination. Some only optimise the trajectory/route for a particular flight-phase such as ground, take off, land down, climb, cruise, or descent phase. Study is categorized by optimisation phase as follows. Most study are for cruise phase as this phase is the longest phase and it can reduce significantly the amount of time, fuel, and emissions if the optimal trajectory or route are found. Algorithms can be developed to find optimal time, fuel etc in cruise phase taking into account wind, weather, and air traffic conditions including algorithms for route optimisation [98, 199, 133] and trajectory optimisation [143]. Take-off and land down phases also receive special concern where algorithms are designed to avoid and minimize the impact of windshear as windshear at these phases is a threat to the aircraft safety. Most study in this phase is for trajectory optimisation [157, 120, 124, 125, 123]. Study on descent phase mainly relates to Continuous Descent Approach (CDA) or Optimised Profile Descent (OPD). This approach can help to reduce significantly fuel, noise and emissions [5]. CDA has been implemented in many airports. The climb performance of an aircraft is an important design requirement for establishing trajectories to reach a specified altitude and airspeed after takeoff in some optimal manner. A number of study has been carried out to minimize fuel consumption or time in climb phase [132, 155]. There is little study, which optimise aircraft trajectory/ route from origin to destination. Smiths Industries (SI) has developed software to compute optimal fuel/time flight profiles and accurately predicting flight trajectories from origin and destination [162]. However the detailed methodology is not provided. In this thesis we propose algorithms to find user preferred routes for flights from origins to destinations.

(d) Category by Objectives

ATRO may consider different objectives which may be minimal time, noise, fuel, emissions, threat, and discomfort. Most study on military and unmanned aerial vehicles consider obstacle avoidance which may be terrain, radar avoidance or bad weather avoidance [149, 19, 96, 146]. The objective of these study is to minimize threat of radar, weather, and terrain to aircraft. The objective of noise is mainly concerned around airport area because of the growth of air traffic demand and urbanization around airports [192, 5]. Many airports need to deal with the impact of noise and emissions on communities around the airports. Time optimal trajectory can reduce fuel, emissions and air traffic controller workload. As indicated in [98], minimizing time in route phase is also to minimize fuel. In [98], time travelled is optimised by flying aircraft in optimal wind route. Air traffic flow management by ground holding is also an efficient method to reduce the time of the aircraft in the air, as ground holding can help to avoid predicted air-borne delays [22]. There have been a lot of work to study and minimize the impact of aviation emissions [172, 31, 203, 76, 194, 173, 5]. These work may directly minimize emissions produced or indirectly through minimize fuel burn. A recent study [194] reports that persistent contrails may have a three to four times greater effect on the climate than CO2 emissions. However, the complete effect of persistent contrails on climate change is still not known as they have both negative and positive effects and the resulting net effect is uncertain. In [173] an algorithm is developed to calculate wind optimal trajectories for aircraft while avoiding the regions of airspace that facilitate persistent contrails formation. The trade-off between persistent contrails formation and additional fuel consumption is investigated.

2.3.3.3 Summary

In summary, there have been a lot of study for solving aircraft trajectory/route optimisation. However there are few dealing with multiple objectives, and resolving the problem with multiple aircraft taking into account aircraft models, and weather information. In this thesis we will solve this kind of problem with innovative and efficient methods in both discrete search space and continuous search space.

2.4 User Preferred Routes

2.4.1 Types and Nature of Users in Air Traffic Management

User Preferred Routing today involves a number of users who are directly responsible for or may be not directly responsible for the operation of commercial aircrafts. They are the Air Navigation Service Providers (ANSPs), the Airline Operation Centers (AOC), the Pilots, Civil Aviation Authorities (CAA), and Public. These users may have different preferences in UPR planning. When an airline does UPR planning, it considers all these preferences and figure out a suitable preference (the preference may be for timely, economical, environmental, or passenger comfort objective).

The details of roles and preferences of each user group are presented in following sections.

2.4.1.1 Air Navigation Service Providers

ANSPs are in charge of avoiding collisions, avoiding system overflow through balancing the capacity and demand of the airspace, and providing a smooth traffic flow, minimizing cost related to congestion (delay, additional distance, non optimum flight level), providing information about weather, navigation, and other relevant information to pilots. This work is usually called Air Traffic Control. The air traffic control includes three segments: En route ATC, Terminal Area ATC, and Airport ATC.

The en-route ATC has responsibility for the controlled airspace between airports. The terminal area (the airspace around airport) is managed by a local Terminal Area ATC in order to deal with the high-density traffic in this area. The Terminal Area ATC works with the arrival and departure routes. However the actual arrivals and departures from airports (the airspace close to the runways and on ground) are handled by Airport ATC.

Air Traffic Controllers need to interact with specially trained personnel, advanced technical equipment and precise, well proven work methods. They pass instructions and authorizations to the pilots, maintain contact with them, and control the position of an aircraft in the airspace. They also need to assure a safe separation between aircraft.

2.4.1.2 Air Safety Authorities

The Civil Aviation Authority (CAA) is the national organization for governing civil aviation in a country. CAA is responsible for providing the community with a safe, cost-effective and environmental friendly civil airspace. They develop national regulations based on international regulations by International Civil Aviation Organization (ICAO) for the national airspace stakeholders such as air navigation service providers, airline, pilots etc. They provide authorization and certificates to airlines, aircraft and crew. They supervise the implementation of regulations by the airspace stakeholders. They publish airspace chart and procedures for the standard use of the national airspace. They carry out studies relating to the security level of the airspace and the aviation environmental impact.

CAA's preferences to UPR include safety, fuel, emissions and passenger comfort. CAA is responsible for strategy in air traffic rather than participating in flight planning. CAA direct users in UPR planning through general policy and regulations. Therefore this thesis does not consider CAA as a user in UPR planning.

2.4.1.3 Air Lines

Airline Operation Center (AOC)

The AOC is responsible for the safe and efficient operation of flights of the airline, following legal regulations. The AOC includes a number of entities such as dispatch, flight crews, ground crews, gate managers, and others. They are coordinated within the AOC for flight operations. The AOC is also in coordination with meteorology, engineering, crew and route planning staff.

At large airlines, all the planning process for the flight is implemented by dispatchers. Primarily dispatchers are in charge of planning and controlling flights safely and legally. They coordinate planning for a specific flight and its impact on the airline's overall schedule. Dispatchers are responsible for flight preparations, and monitor the flights status and weather conditions during the flight. In preparing a flight plan, the fuel required, payload, route to be flown, and alternate airports for each flight are provided.

The concern of aircraft dispatchers in making decision is whether or not a flight is safely operated. Items taken into account are the deferred Minimum Equipment List (MEL) or Configuration Deviation List (CDL) items (which identify particular equipment inoperative, subject to specified conditions or any external parts of an aircraft type which may be missing at the commencement of a flight, and which contains, where necessary, any information on associated operating limitations and performance correction [87]), aging aircraft restrictions, and the required technical and operational support at the departure, destination and alternate airports including fuel, gates, ground power unit (which is a vehicle capable of supplying power to aircraft parked on the ground), and stairs etc. The role of aircraft dispatcher is expanded to deal with the additional coordination when irregular operations and emergencies happen.

In smaller airlines, the pilots take many functions of the dispatcher. The pilots can access to all the essential weather data and aircraft status and then do all the flight planning.

AOC's preferences to UPR include safety, fuel, emissions and passenger comfort. Dispatchers work in airline operation centers and are responsible for strategic flight planning.

Pilots

Flight planning can be a task of the dispatchers or the flight crew. During the actual flight, through communication systems the pilots have access to data to support flight planning safely, economically, and environmentally. The pilots can operate the flight as planned and also can even optimise it, taking into account the updated weather conditions and flight status.

The details of the trajectory are usually kept on the aircraft. Air traffic controllers only know aircraft flying from an origin to a destination through certain points at certain times. They control aircraft (especially in approach and landing phase) by informing the pilots to slow down or to achieve a certain heading at a certain altitude so that aircraft flies some traffic pattern or a flow to the airport. The controllers don't know the optimum aircraft performance (which depends on aircraft model, weight, weather etc.), so their commands often don't fly the aircraft the optimal user preferred trajectory. On the other hand, the pilots don't have a clear picture about the objectives and restrictions from the controller, so they can't use updated information to find optimal trajectory. What the pilots do is trying to figure out the traffic situation around by using means of information especially through the conversation with the controllers and then do find out what flight path the controllers can expect.

Pilot's preferences to UPR include: safety, fuel, emissions and passenger comfort.

2.4.1.4 Public

Safety is always the first expectation from the public (particularly passengers), because ultimately passengers will suffer the consequences of an unsafe system, or a safe system that is operated unsafely. In addition, passengers concern about flight schedules. They may have some favorable departure times, and wish to depart and arrive on time. They may also wish to have sufficient choices of flights and itineraries, cheap and comfortable flights. Another expectation from the public is national security and environment. Particularly the public in the vicinity of airports expects less noise pollution. These expectations are responded in different ways by the airspace user community, e.g. by airlines in their flight paths, schedules and fare structures.

2.4.2 Variants of User Preferred Routes

User Preferred Routes is a link in the progress towards User Preferred Trajectories (UPTs). In [138] Paglione mentioned that "While the initiatives in Europe and the United States were still just discussions among aviation stake-holders, Australia embarked on a world first initiative to develop an ATM Strategic Plan as early as 1999. Based on a collaborative approach with User Preferred Trajectories as the ultimate goal; the ATM Strategic Plan established a framework that enables Australia to keep at the forefront of the Communications, Navigation, and Surveillance (CNS) systems and ATM development and its associated benefits". According to the plan, the transition towards UPTs will begin with flex tracks, followed by User Preferred Routes, and User Preferred Trajectories [14]. In Europe and the United

States UPRs are determined as the base for 4D Trajectory Management as presented in Section 2.1.3.

There have been research on User Preferred Routes and User Preferred Trajectories carried out by FAA and Eurocontrol [29]. Since late 1995, the Eurocontrol Experimental Centre (EEC) launched a project named FREER, for Free-Route Experimental Encounter Resolution to investigate the feasibility of the concept following which ATC functions could be transferred to the flight deck to allow more freedom of movement to airspace users, and to support the implementation of Free Flight, Free-Route and User-Preferred Route concepts [51]. NASA Ames Research Center is developing decision support tools for air traffic controllers to improve the efficiency and capacity of the National Airspace System. The goal is to provide technology and procedures that result in the highest possible level of user preferred routing whenever possible with efficient traffic management when necessary [115]. As presented in Section 2.3.3.2, there have been a lot of study on aircraft trajectory and route optimisation in general, but there is little study dealing with multiple objectives, especially proposing methods for solving UPR or UPT problems.

Following sections describe Flex tracks, User Preferred Routes, and User Preferred Trajectories.

2.4.2.1 Flex Tracks

As defined by ASTRA (The Australian Strategic Air Traffic Management Group), "Flex Tracks are the non-fixed air traffic services routes calculated and promulgated daily by the air navigation service provider to provide the most efficient operational flight conditions between specific city pairs, taking advantage of prevailing wind conditions. The Flex Track will commence and terminate at designated waypoints to allow for transition to the fixed route network for arrivals and departures" [14].

Flex Tracks will save fuel and benefit the environment. It is a block towards achieving user preferred routes [14].

2.4.2.2 User Preferred Routes

User preferred routes are defined as "the routes generated and flown by an airspace user that reflect the optimum point-to-point flight path subject to overriding system constraints with links to waypoints from/to structured routes at both ends. They provide further flexibility, efficiency and cost effectiveness to airspace users compared to Flex Tracks that are pre-determined and agreed by a service provider." [14]. A route can either be a 2D route (with lateral containment), or 3D routes (with lateral and vertical containment).

With the support of appropriate conflict detection tools, user preferred routes will enable airspace users to achieve specific operational/businees objectives (e.g. minimum flight time, fuel, or discomfort) subject to overriding ATM system constraints. [14].

The gradual maturity of the operational concept, procedures, air traffic controller automation tool proficiency and enhanced ATC/AOC/flight crew collaboration will make user preferred routes available progressively. Flex Tracks will be gradually removed as user preferred routes are fully in practice [14].

2.4.2.3 User Preferred Trajectories

The User Preferred Trajectory can be considered as the extension of User Preferred Route in time dimension. As defined by ASTRA, "The User Preferred Trajectory (UPT) is normally determined in terms of best business outcome, which considers time, fuel burn, and/or systemic cost. The UPT may take into account known and/or predicted environmental constraints and is described in terms of 4D coordinated and a statement of operational preference. Operational preference indicates the priority for application" [13]. The four dimensions are three spatial dimensions, plus time. A UPT begins when an aircraft is established at the gate for the purpose of the flight until the aircraft has arrived the destination arrival gate. UPTs therefore contain en-route, terminal area (TMA) and ground phases of the trajectory.

2.5 Chapter Summary

Though User Preferred Routes (UPRs) is described in literature, there is little study to resolve UPR problem. Furthermore, there has been little study to investigate the impact of UPR on the environment, delays, safety, etc. This motivates us to study the User Preferred Routing problem in aspects of formulating User Preferred Problem more comprehensively, developing methods to resolve the UPR problem properly and efficiently, and analysing the impact of UPR routes (or evaluating UPR concept) in this thesis. Specifically the UPR problem takes aircraft models, user preferences, and weather data such as wind, and hazards as input and multiple objectives as output for finding user preferred routes for large number of aircraft in continuous environment. The impacts of UPR routes on environments and departure time deviation are analysed.

Particularly we will find answers for following research questions in this thesis.

- What is an appropriate simulation environment for User Preferred Routing?
- What are efficient designs for GA and LCS for finding UPR?
- What is the difference between black box (GA) and white box (LCS) approaches in finding UPR?
- What is the efficient method to detect and resolve conflicts between flights?
- What is a systematic use of UPR methods to take advantages of each method in a specific routing circumstance?
- How is a real time system for aviation emission computation, inventory development and analysis designed?
- What is the difference of environmental impact between UPR routes optimised only vertically, only horizontally, and both vertically and horizontally?
- What is the impact on departure time deviation when UPR routes are applied?

Chapter 3

The Simulation and Evaluation Environment

This chapter is partially based on following publications:

1. V. Bui, V. V. Pham, A. W. Iorio, J. Tang, S. Alam, and H. A. Abbass. Bio-inspired robotics for air traffic weather information management. Transactions of the Institute of Measurement and Control, pp. 1-27, ISSN: 0142-3312, 2010

3.1 Overview

The problems to find 3-D (latitude, longitude, altitude) UPR routes are described first. The bad weather model and weather data processing, which provide the input data for determining UPR routes, are presented next. Then the simulation environment is presented. The simulation is used to evaluate and provide 4-D trajectories with high resolution. UPR routes are then obtained by removing time dimension from the 4-D trajectories. The conflict detection and resolution method is presented next, and is applied to detect and resolve conflicts among UPR flights and with other non-UPR flights, when the 4-D trajectories of flights are provided by algorithms and through the simulation environment. Finally we present experimental framework to investigate algorithms for finding 3-D UPR routes.

3.2 Problem Formulation

A user at the current time wishes to find 3-D UPR for a set of flights in a time-frame into the future. For example at 6am in a day, he wishes to find 3-D UPR for a set of flights whose given departure time is from 2pm to 3pm. While finding the routes for the set of UPR flights, he also needs to consider how to resolve conflict among UPR flights and other flights in the airspace (which are called non-UPR flights). The non-UPR flights may be active or set to take off. Both types of these non-UPR flights are given flight plans.

In this thesis, we propose a framework and algorithms to find 3-D (latitude, longitude, altitude) user preferred routes for flights in a constrained environment. We study two problems separately. Flights in Problem 1 are pre-assigned 2-D (latitude, longitude) routes, while those in Problem 2 are pre-assigned cruising altitudes. By studying two problems separately we can see how each dimension can be optimised. Further the methodologies proposed for the two problems can then be combined to optimise both horizontally and vertically. Though we can simulate a 3-D route to obtain the 4-D trajectory by using aircraft speed profile from BADA database, we consider this problem as 3-D User Preferred Routing as the speed of the aircraft, which is used to determine the time dimension of the trajectory, is not optimised.

3.2.1 Problem 1

Problem 1 is given a set of UPR flights whose routes are user preferred, a time frame that contains departure times of the UPR flights, a set of non-UPR flights in the airspace that are predicted to be time-overlapping with UPR flights, and weather environment with wind and bad weather information. Each flight has a origin, an destination, given departure time, an aircraft type, a route, and a utility (which represents the preference between the objectives: time, and comfort). The task is to find 3-D user preferred routes for UPR flights, so that UPR flights can avoid conflict with each other and with non-UPR flights, and can minimize time and discomfort. Conflicts between UPR flights and other flights are resolved by shifting the departure times of the UPR flights.

This problem is named as GRP (Given Route Problem).

The problem is formulated mathematically as follows.

Input Data

- A departure time frame $[T_1, T_2]$
- A set of n_1 UPR flights: f_0,... f_{n_1-1} which depart in the time frame $[T_1, T_2]$ and which need to find user preferred routes. Each UPR flight f_i has the following information.
  - $Orig_i$ is the origin of f_i
  - $Dest_i$ is the destination of f_i
  - DT_i is the departure time of f_i in the time frame $[T_1, T_2]$.
  - AC_i is the aircraft type of f_i
  - $Route_i$ is the route which is a series of m_i 2-D (latitude, longitude) way-points: $WP_{i,0}$, $WP_{i,1}$,... WP_{i,m_i}
  - u_i is the utility of f_i. The utility of a flight ranges from 0 to 1. If the utility is high, minimizing the time travelled is more preferred, otherwise minimizing the discomfort is more preferred. For example, for Cargo flights that have no passengers, an airline is likely to prefer a quicker route than a comfortable one.
- A set of n_2 non-UPR flights: f'_0,... f'_{n_2-1}. A non-UPR flight f'_i has the following information
  - 4-D predicted trajectories which are generated from its flight plan.
  - Its travelled time from the departure time to the arrival time must overlap with the period $[T_1, T_3]$ where T_3 is large enough so that all UPR flights arrive to their destination before T_3. The travelled time of the flight can be received from the 4-D predicted trajectories.

- A 3-D wind cell grid $WindAtm_{i,j,k}$, where i, j, k are the latitude, longitude, and altitude indices. A wind cell $WindAtm_{i,j,k}$ has the following information
  - *Speed* is the speed of wind at cell i, j, k. Wind speed is measured in knots
  - *Direction* is the direction of wind at cell i, j, k. Wind direction is measured in degree from 0 to 360° (from North in clockwise direction).
- A 3-D bad weather cell grid $BWC_{i,j,k}$, where i, j, k are the latitude, longitude, and altitude indices. A bad weather cell $BWC_{i,j,k}$ has the following information
  - BWL is the bad weather level at cell i, j, k. Bad weather level can take an integer value between 0 and 5 which correspond to *none*, *low*, $medium - low$, *medium*, $medium - high$, or *high* respectively.
  - *Down* is the lower altitude of the bad weather at cell i, j, k
  - *Up* is the upper altitude of the bad weather at cell i, j, k

Find
3-D user preferred routes from origin to destination through the constrained environment for UPR flights so that they can satisfy the constraint and minimize the objective below.

Constraint
- UPR flights must avoid conflict with each other and with the non-UPR flights.
- All flights must operate within their specific aircraft allowed performance parameters.

Objective
For UPR flights, minimize the following objective

$$\sum_{i=0}^{n_1-1} \sum_{j=0}^{ns_i-1} (u_i * time_{i,j} + (1 - u_i) * BWL_{i,j} * time_{i,j}) \tag{3.1}$$

where ns_i is the number of segments of the 4-D trajectory of flight f_i, $time_{i,j}$ is the time travelled by flight f_i through the segment j of the flight, $BWL_{i,j}$ is the bad weather level of the segment. Here we assume that the segment is small enough that the bad weather level is constant in the whole segment.

As the time travelled of a flight is usually proportional to fuel burn as well as emissions [98], minimizing time travelled is an indicator to minimize fuel and emissions.

3.2.2 Problem 2

Problem 2 is formulated identically to Problem 1, except for one thing. The input data for a UPR flight f_i does not include $Route_i$ (a set of 2-D waypoints); instead, it includes $CruiseAlt_i$, the cruising altitude of f_i.

This problem is named as GAP (Given cruising Altitude Problem).

3.3 Bad weather model

A variety of bad weather phenomena can affect aircraft, such as wind shear, turbulence, icing, atmospheric electricity, stratospheric ozone, volcanic ash, condensation trail, whiteout, low-level inversions, and thunderstorm [112]. Wind shear can cause turbulence and large airspeed fluctuations particularly during the takeoff and landing phases of flight. Turbulence results in velocity fluctuations in a chaotic manner. The fluctuations found within the atmosphere are often weak and barely noticeable in flight. However, turbulence of high magnitude occasionally causes aircraft damage or passenger and crew injury. Icing is possible in clouds or rain at temperature near or below 0°C. It can quickly become a factor and threaten the safety of flight. Atmospheric electricity can be lighting and static discharges which are the concern for pilots. The air with high ozone concentration from stratosphere may create a hazard of flights at and above the tropopause. Volcanic ash can speed around the world and remain in the stratosphere for months, which can lead to the interruption of air traffic. Condensation trail (clound-like streamer) that forms behind an aircraft flying in clear, cold, humid air can reduce visibility at a flight level. Whiteout (Snow-covered regions at high altitudes) can present some special visibility problems because of the reflection of light by snow-covered surfaces or the diffusion of light from the sun by the cloud layer. Strong low-level inversions are common in snow-covered regions in the winter. An aircraft climbing out under such conditions experiences a marked decrease in climb performance due to higher temperatures at the top of the inversion. The dangers a thunderstorm may cause are loss of control, loss of power, and damage by hail, lightning, or extreme turbulence. However these dangers can all be foreseen and precautions can be taken to minimize their effect [74].

In this thesis, we abstract the impact of these different weather phenomena by weather cells with sizes of 1° x 1° x 1,000 feet. The intensity of a bad weather cell is assumed to be constant in the whole cell. Flights can pass through bad weather cells, but they will create discomfort. In reality a wether cell can scale from 2 to 200 km [114] and can be from the ground to 60,000 feet [74] and the weather cells are changing continually. In this thesis we simplify these weather facts to study UPR problems. However the proposed methodologies can be adapted to deal with the real weather data by replacing the 3-D weather grid with 4-D weather grid. For example, in order to detect a point in a bad weather area in the 3-D bad weather cell grid, its position is mapped to 3-D grid. The detection in 4-D grid the point is almost the same where the point with estimated time of arrival is mapped to the 4-D grid to see if the grid cell containing the point is the bad weather area.

3.4 Weather Information System

In this thesis we designed a system to process and integrate aviation weather. The research work is published in [32]. The resulting wind data is then used in User Preferred Routing.

A weather monitoring system typically consists of sensors (both airborne as well as ground) located in or near the terminal area as well as local and regional forecast information. The main types of information generated by aviation weather

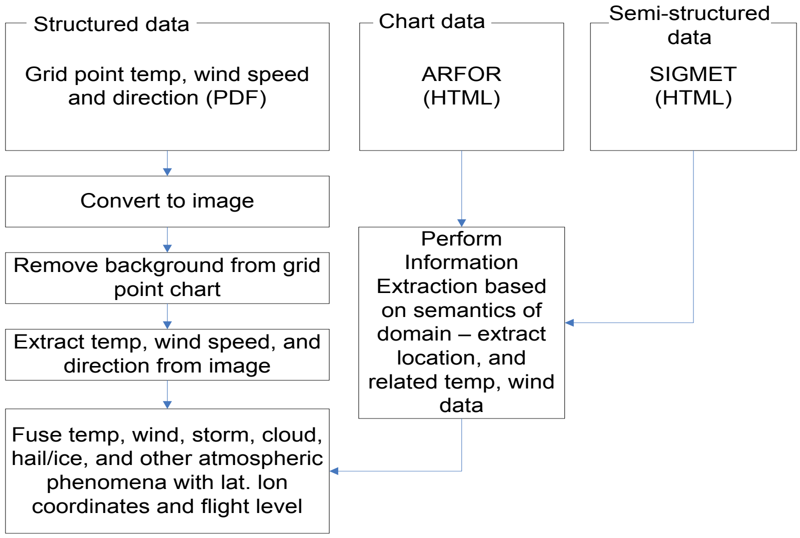


Figure 3.1: Process flow diagram for weather data processing.

monitoring systems are the SIGMET (Significant Meteorological Hazard Warning), AIRMET (Airmen's Meteorological Warning), TTF (Trend Type Forecast), TAF (Terminal Aerodrome Forecast), ARFOR (Low-level Area Forecasts), Area QNH (air pressure), SIGWX (Significant Weather Charts) and grid-point wind, and temperature forecasts [112].

The weather information system can retrieve and process SIGMET, ARFOR, and numerical grid-point wind, and temperature data. This data consists of semi-structured and structured data. The system accesses the BoM (Bureau of Meteorology, Australia) website, downloads the HTML messages containing SIGMETs, and parses them. The SIGMET parser extracts structured data from the SIGMET, identifying the region, weather phenomena, flight level, and time of the event. This data is then stored in a database for retrieval and future updates.

The interpretation of region information in a SIGMET is typically reported as vertices of a polygon in latitude and longitude coordinates. Sometimes this data is presented as a mix of waypoint codes, or latitude and longitude coordinates. For ATCs that may look at more than one sector in a day, depending on traffic load, this can be burdensome as the ATCs have to have familiarity with the waypoints in each region.

The weather information system automates the extraction of the waypoint codes and converts this data to latitude and longitude coordinates, and can provide the most current SIGMET within a specified time window. Furthermore, the system extracts numerical grid-point wind, and temperature data from different sources. In addition to the SIGMET and numerical grid-point wind, and temperature data, the weather information system also downloads and parses the low-level ARFOR (Area-forecast) data. Figure 3.1 illustrates the process flow for weather data processing.

3.4.1 Neural Networks for Wind and Temperature Image Processing

Wind and temperature data in Australia is generated every 6 hours by the Bureau of Metrology in image format. This data is unavailable in a textual format so this image needs to be converted into a text format for further machine processing. Each image (map of Australian airspace) is divided into grid cells based on latitude and longitude. The size of each cell is 5° x 5°. Each cell has wind and temperature information for 9 different altitudes (5000, 10000, 14000, 18500, 23500, 30000, 34000, 38500, and 44500 feet).

Figure 3.2 is an example of a wind and temperature map from 5000 to 23500 feet. The wind and temperature information is in the format of $ddffftTT$. dd are the two digits for wind direction with the unit of 10 degrees. These two digits present approximately 1/10 of the real wind direction. The value made from the two digits is from 0 to 36. This means the real wind direction is 10 multiplied by the integer value made of the two digits. *fff* are the three digits for wind speed with the unit of knots. t is the sign of temperature. It is "+" if the temperature is positive and it is "-" if the temperature is negative. TT are the two digits for the temperature with the unit of Celsius degree.

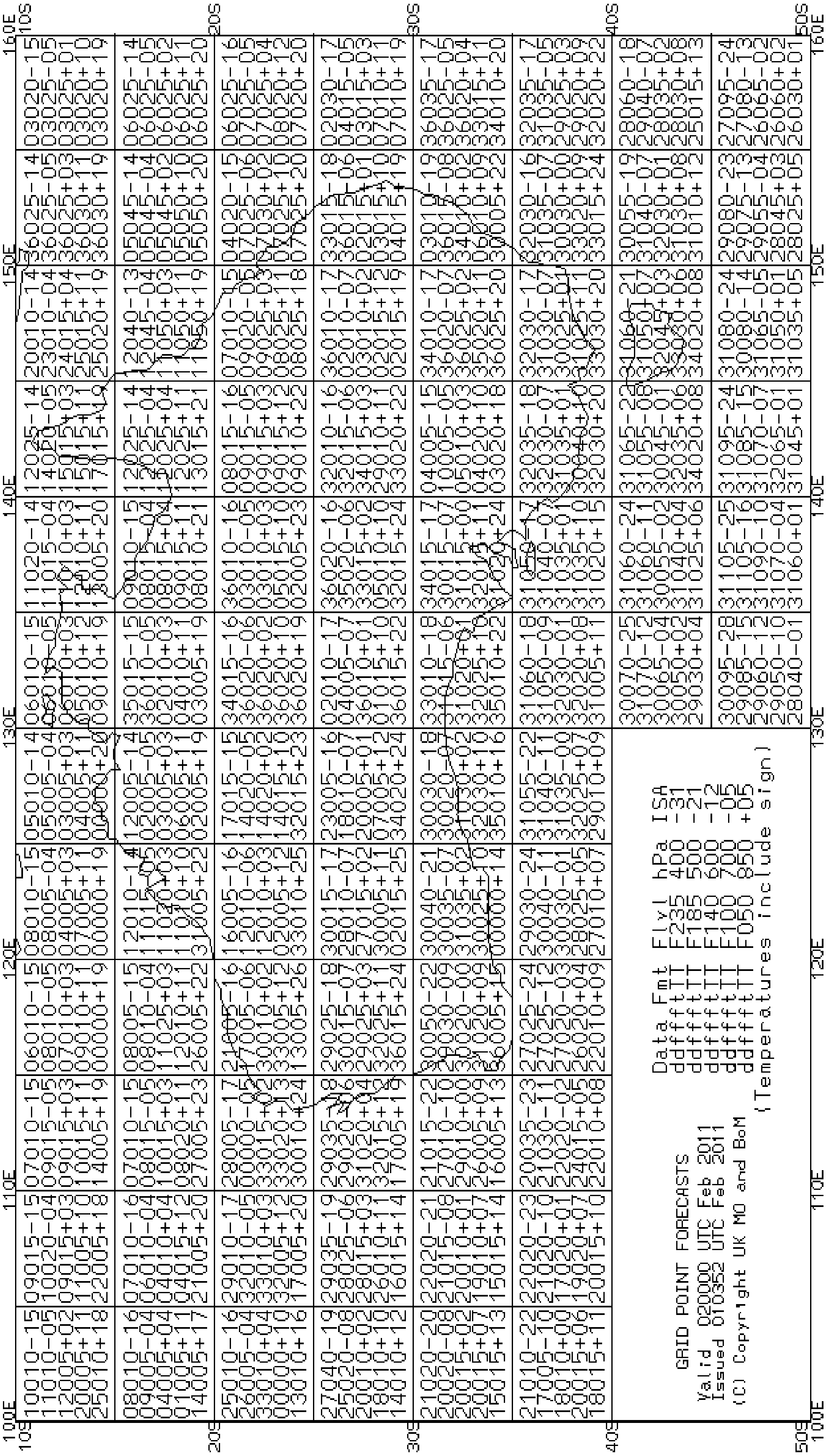


Figure 3.2: Wind and temperature image file.

Wind and temperature data from 30000 to 44500 feet are received from another similar wind and temperature map. The information in this wind and temperature map is in the format of $ddfffTT$ without temperature sign as all the temperatures are negative.

In general, the problem is seen as a pattern recognition problem where the patterns to identify are 10 digits from 0 to 9 and the two temperature signs ("+" and "-"). The problem of converting this image into a system readable text file is dealt with in two stages; in the first stage we filter out the background clutter (map of Australia) and localize and segment the image based on the location of individual digits and temperature signs. In the second stage an off-line trained neural network is used to identify the digits in wind and temperature image files. The temperature signs are simply recognized by the difference between the two signs. Sign "+" has pixels in different rows while all pixels of sign "-" are in the same row.

3.4.1.1 Pre-processing of Wind and Temperature Images

Before we can train a neural network to identify digits in wind and temperature images, several preprocessing steps need to be performed in order to generate the cases used for training. The map of Australia overlaps a number of digits and temperature signs and the size (width and height) of one digit sometimes varies from one to another. Utilizing several wind and temperature image files the unchanging background pixels associated with the grid and map of Australia are identified. Following this, we acquire the digits and temperature signs from each image by subtracting the background image and drawing another automatically constructed grid on the image in order to segment the digits and temperature signs. The segmentation of the grid is generated by iterating over every pixel row and column of the image; the pixels that become associated with the segmentation grid are the rows and columns with white pixels only. If there are several consecutive white rows and columns we consider them as one. The grid then allows us to segment the individual digits and temperature signs with high precision. The temperature signs are simply recognized as presented above. The segmented set of digits is then used in the second stage

Table 3.1: Parameter for two neural network models.

Parameters	Values
Network type	Feed-forward backprop
Hidden layer	Transfer function is *tansig*.
Output layer	Transfer function is *tansig*.
Goal	0
Epochs	100
Momentum constant	0.95
Learning rate	0.01

where a neural network is used to recognize digits in new wind and temperature images. The size of the training set used by the neural network is 317, where the number of templates for each digit from 0 to 9 is 29, 16, 25, 52, 19, 31, 50, 9, 30, and 56 respectively. These numbers were used to reflect the variations associated with the font used in the image for the digits.

3.4.1.2 Training the Neural Network

We used a set of neural networks that are trained separately to recognize each digit. Each trained network has one hidden layer with 10 units and one output. The output reports true if an image of a digit is matched, otherwise it reports false. The architecture of the 10 neural networks is presented in Figure 3.3. The neural network parameters are reported in Table 3.1. As shown in the table:

- The network type we use is feed-forward backpropagation [77]. In a feed-forward neural network, neurons are only connected forward. Each layer of the neural network contains connections to the next layer (for example, from the input to the hidden layer), but there are no connections back. Backpropagation is a form of supervised training. When using a supervised training method, the network must be presented with both sample inputs and predicted outputs. The predicted outputs are compared with the actual outputs corresponding to a given input. On the basis of the predicted outputs, the backpropagation training algorithm then takes a calculated error and alter the weights of the layers backwards from the output layer to the input layer.

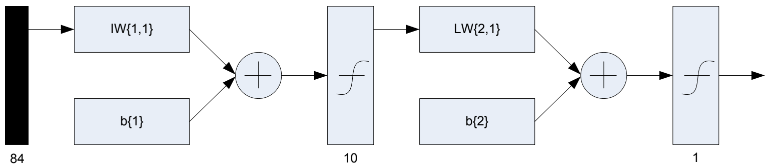


Figure 3.3: The architecture of the 10 neural networks with one output.

- The transfer function used in the network is the *tansig* function. This function transfers the weighted sum of the input nodes to the output as shown in Equation 3.2

$$y = f(a) = f(\sum_{i=0}^{n} x_i w_i) \tag{3.2}$$

where $x_i (i = 1, ..., n)$ are elements of an input vector x, x_0 is a threshold input, w_i is a weight from an input i to a neuron, and $f(a)$ is the tansig function given in Equation 3.3.

$$f(a) = tansig(a) = \frac{2}{1 + e^{-2*a}} - 1 \tag{3.3}$$

- Goal, Epochs, and Momentum constant are training parameters. Training stops when a maximum number of epochs occurs or the performance goal is met. Momentum constant (MC) determines the amount of momentum. MC is ranged from 0 (no momentum) to values close to 1 (lots of momentum). When the momentum constant is 1, the network is completely insensitive to the local gradient and, hence, does not learn properly. The learning rate for both input weights and biases is 0.01. This constant is used in error backpropagation learning and other artificial neural network learning algorithms to control the speed of learning.

The training set for each neural network consists of the set of template images for all 10 digits, even though a neural network is trained to recognize a specific digit only. This is necessary so that each network can see negative as well as positive examples in order to correctly distinguish between digits. The 84-bit strings used in

0011110	0111100	0011100	0111000	0011100
0100001	1000010	0100010	1000100	0100010
1000001	1000010	1000001	1000010	0100001
1000001	1000001	1000001	1000010	0100001
1000001	1000001	1000001	1000010	0100001
1000001	1000001	1000001	1000010	0100001
1000001	1000001	1000001	1000010	0100001
1000001	1000001	1000001	1000010	0100001
1000001	1000001	0100010	0100100	0100001
0100010	1000001	0011100	0111000	0010010
0011110	0100010	0000000	0000000	0011100
0000000	0111100	0000000	0000000	0000000

Figure 3.4: Five samples of digit 0.

training originate from the binary matrix representing the black and white image of a digit, where the width of the matrix is 7 and the height is 12. These digit images are extracted from Figure 3.2 by the method, which is described in Section 3.4.1.1. In a binary matrix representing a digit image, bit 0 represents a white pixel, and bit 1 represents a black pixel. In Figure 3.4, examples of this binary matrix are presented for digit 0.

Figure 3.5 shows the performance of the training process for the neural network to recognize digit 0. It reaches the goal with the performance very close to 0.

3.4.1.3 Using Neural Networks for Classifying Wind and Temperature Images

Once the neural network is trained it can then be used to process new grid-point wind and temperature data images. When a new image arrives for processing, we perform the same preliminary preprocessing steps that were used before training. After this, the segmented digits can be recognized using the trained neural networks. In order to recognize one new image of a digit, all ten trained neural networks are used. If only one of the neural networks has the output of true during classification, the digit that this neural network is responsible for identifying is deemed to be the digit corresponding to the new image, otherwise the new image is deemed to be unrecognised. The neural networks are applied to each digit in the image until there

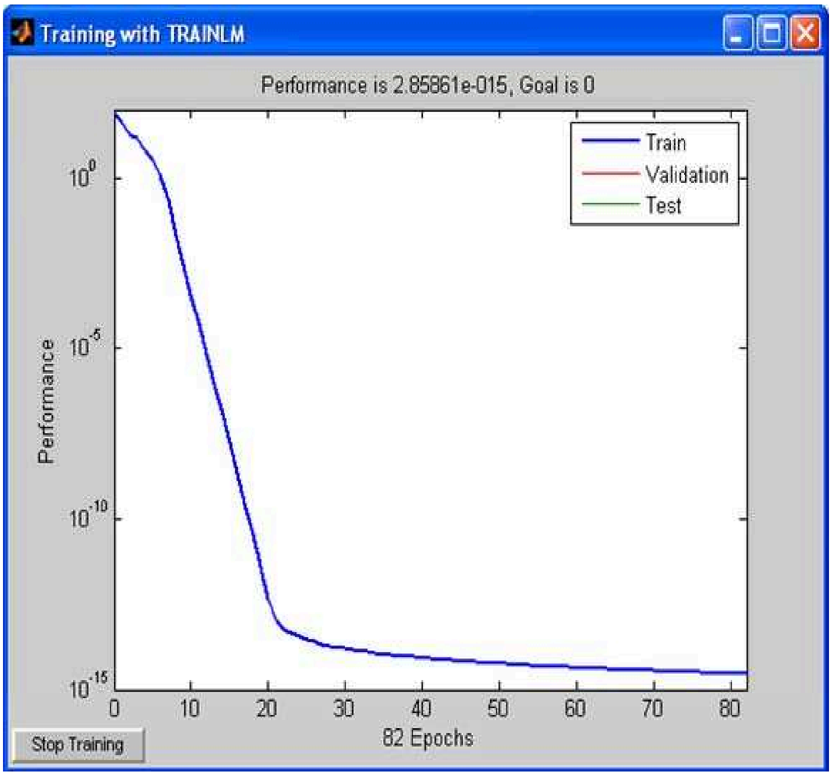


Figure 3.5: Training performance.

are no more digits to classify. The wind, and temperature information extracted from the image is then integrated to give a complete picture. The graphical representation of this meta-data generated by processing weather, and wind information is then visualized as given in Figure 3.6.

We test the trained neural networks on 10 different image files of wind data. The image files are converted to text files by using these networks. The results show that the accuracy obtained is 100%. Table 3.2 is an example of the text file of wind and temperature, converted from the image file in Figure 3.2. The cells in the table with only 0 digits correspond to the description area of the wind and temperature map.

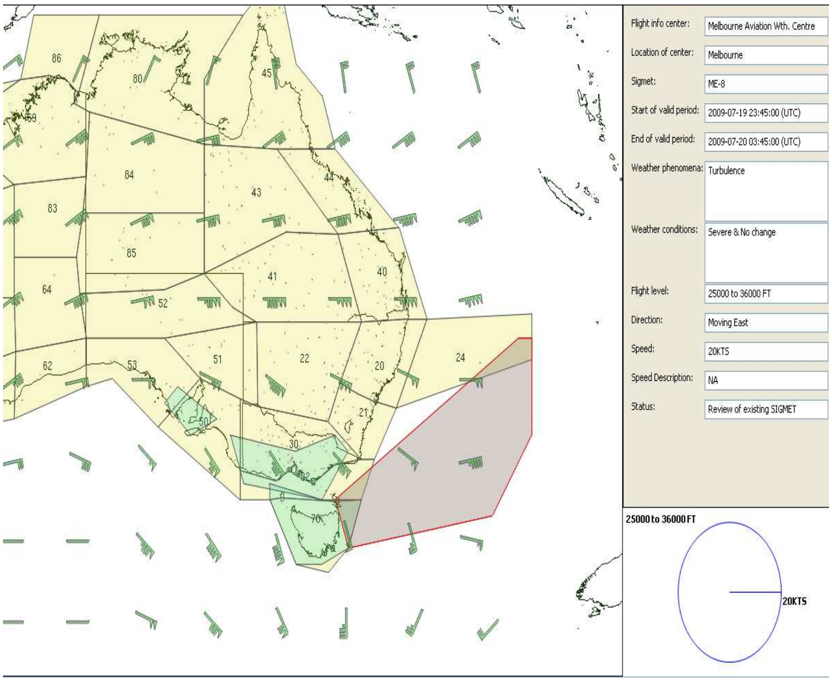


Figure 3.6: Graphical output of meta-data generated by processing weather and wind information.

10010-15	09015-15	07010-15	06010-15	08010-15	05010-14	06010-15	11020-14	12025-14	20010-14	36025-14	03020-15
11010-05	10020-04	09015-05	08010-05	08005-04	05005-04	06010-05	11015-04	14020-05	23010-04	36025-05	03025-05
12005+02	09015+03	09015+03	07010+03	04005+03	03005+03	06010+03	11010+03	15015+03	24015+04	36025+03	03025+01
20005+11	11005+10	10015+11	09010+11	07005+11	04005+11	05010+12	12010+11	15015+11	25015+11	36030+11	03020+10
25010+18	22005+18	14005+19	00000+19	00000+19	00000+20	09010+19	13005+20	17015+19	25020+19	36030+19	03020+19
08010-16	07010-16	07010-15	08005-15	12010-14	12005-14	35015-15	09010-15	12025-14	12040-13	05045-14	06025-14
09005-04	06010-04	08015-05	08010-04	11015-05	02005-05	36010-05	08015-04	12025-04	12045-04	05045-04	06025-05
04005+04	04010+04	10015+03	11025+03	11020+03	03005+03	02010+03	08015+04	12025+04	11050+03	05045+02	06025+02
01005+11	04015+12	08020+11	12010+11	11010+10	06005+11	04010+11	09015+11	12025+11	11050+11	05050+10	06025+11
14005+17	21005+20	27005+23	26005+22	31005+22	02005+19	03005+19	08010+21	13015+21	11050+19	05050+20	06025+20
25010-16	29010-17	28005-17	21005-16	16005-16	17015-15	34015-16	36010-16	08015-16	07010-15	04020-15	06025-16
26005-04	32010-05	00000-05	17005-06	12010-06	14020-05	36020-06	03010-05	09015-05	09020-05	07025-06	07025-05
33010+04	33010+03	33015+02	30010+02	12010+03	14020+02	36020+02	03010+03	09015+03	09025+03	07030+02	07025+04
00000+10	32005+12	33020+13	33005+12	02005+12	14015+10	36020+10	05015+10	09015+12	08025+11	07025+10	08020+12
13010+16	17005+20	30010+24	13005+26	33010+25	32015+23	36020+19	02005+23	09010+22	08025+18	07025+20	07020+20
27040-19	29035-19	29035-18	29025-18	30015-17	23005-16	02010-17	36020-16	32010-16	36010-17	33015-18	02030-17
25020-08	28025-06	29020-06	30015-07	28010-07	18010-07	04005-07	35020-06	32015-06	02010-07	36015-06	04015-05
20010+02	28015+03	31020+04	29025+03	27015+02	20005+01	36010+01	33015+02	34015+03	03010+02	02015+01	03015+03
18010+10	26010+11	32015+12	32025+11	30005+11	07005+12	01015+10	05010+10	29010+11	02015+12	03015+10	07010+11
14010+12	16015+14	17005+19	36015+24	02015+25	34020+24	36015+22	32015+24	33020+22	02015+19	04015+19	07010+19
21020-20	22020-21	21015-22	30050-22	30040-21	30030-18	33010-18	34015-17	04005-15	34010-17	03010-19	36035-17
20020-08	21015-08	27010-10	29030-09	30035-07	30020-07	31015-06	30015-07	10005-05	36020-07	36015-08	36025-05
20015+02	20010+01	29010+00	30020-00	31035+02	31030+02	31020+01	31015+00	05010+03	36025+02	34010+02	36020+04
20015+07	19010+07	26005+09	31020+09	30025+10	32030+10	32025+10	32015+11	04020+10	35020+11	06010+09	33015+11
15015+13	15015+14	16005+13	36005+15	00000+14	35010+16	35010+22	31020+24	03020+18	36025+20	30005+22	34010+20
21010-22	22020-23	20035-23	27025-24	29030-24	31055-22	31060-18	31050-17	32035-18	32030-17	32030-16	32035-17
17005-10	21020-10	21030-11	27025-12	30040-11	31040-11	31050-09	31040-07	32030-07	31030-07	31035-07	31035-05
18010+00	17020+01	22020-02	27020-03	30030-01	31035-00	32030-01	31035+01	31035+01	30025+01	30030+00	29025+03
20015+06	19020+07	24015+05	26020+04	28025+05	32025+07	32020+08	31035+10	30040+10	31025+11	33020+09	29020+09
18015+11	20015+10	22010+08	22010+09	27010+07	29010+09	31005+11	31020+15	32030+20	33030+20	33015+24	32020+22
00000000	00000000	00000000	00000000	00000000	00000000	30070-25	31060-24	31065-22	31060-21	30055-19	28060-18
00000000	00000000	00000000	00000000	00000000	00000000	31070-12	31060-11	31055-08	31050-07	31040-07	29040-07
00000000	00000000	00000000	00000000	00000000	00000000	30065-04	30055-02	30045-01	32045+03	32030+01	28035+02
00000000	00000000	00000000	00000000	00000000	00000000	30050+02	31040+04	32035+06	33030+06	31030+08	28030+08
00000000	00000000	00000000	00000000	00000000	00000000	29030+04	31025+06	34020+08	34020+08	31010+12	25015+13
00000000	00000000	00000000	00000000	00000000	00000000	30095-28	31105-25	31095-24	31080-24	29080-23	27095-24
00000000	00000000	00000000	00000000	00000000	00000000	29085-15	31105-16	31085-15	30080-14	29075-13	27080-13
00000000	00000000	00000000	00000000	00000000	00000000	29060-12	31090-10	31070-07	31065-05	29055-04	26065-02
00000000	00000000	00000000	00000000	00000000	00000000	29050-10	31070-04	32065-01	31050+01	28045+03	26060+02
00000000	00000000	00000000	00000000	00000000	00000000	28040-01	31060+01	31045+01	31035+05	28025+05	26030+01

Table 3.2: Wind and temperature data in text converted from image using trained neural networks.

3.4.2 Wind and Temperature Data Structure and Approximation

Wind and temperature information are real information from processing the Australian wind and temperature map of 5° x 5° cells, as presented in Section 3.4.1. Each cell has wind and temperature information for 9 different altitudes (5000, 10000, 14000, 18500, 23500, 30000, 34000, 38500, and 44500 feet).

A 3-D matrix is used to store wind and temperature information $WindAtm_{i,j,k}$, where i, j, k are the latitude, longitude, and altitude indices. Latitude index is from 0 to 7 corresponding to the latitudes from 12.5S to 47.5S with interval 5°. Longitude index is from 0 to 11 corresponding to the longitudes from 102.5E to 157.5E with interval 5°. Altitude index is from 0 to 8 corresponding to the altitude of 5000, 10000, 14000, 18500, 23500, 30000, 34000, 38500, and 44500 feet.

Each 5° x 5° cell has one tuple including wind direction, wind speed and temperature for a number of flight levels. Therefore, in this thesis, we interpolate tuples to get the weather information in every point in the airspace. Given a point (Lat, Lon, Alt) in a cell which is temporarily called cell_0, call neighboring cells of cell_0 as cell_1, cell_2, ... cell_n (which are the upper, lower, left, right, upper-left, upper-right, lower-left, or lower-right cells if they exist). For each cell, we calculate the tuple at Alt by linear approximation.

If there is one tuple in the cell at Alt, this tuple will be the tuple which needs to be determined. If there is no available tuple in the cell below Alt, the linear approximation will be based on the two tuples above Alt. If there is no available tuple in the cell above Alt, the approximation will be based on the two tuples below Alt. Otherwise it will be based on one tuple below Alt and one tuple above Alt. Assume that the altitudes of the two tuples are Alt_x, and Alt_{x+1}; the wind direction, wind speed and temperature of the two tuples are (d_x, f_x, t_x) and (d_{x+1}, f_{x+1}, t_{x+1}), the 3 values of the tuple (d, f, t) at Alt are calculated by Equation 3.4.

$$\begin{aligned} d &= \frac{d_{x+1} - d_x}{Alt_{x+1} - Alt_x} \times (Alt - Alt_x) + d_x \\ f &= \frac{f_{x+1} - f_x}{Alt_{x+1} - Alt_x} \times (Alt - Alt_x) + f_x \\ t &= \frac{t_{x+1} - t_x}{Alt_{x+1} - Alt_x} \times (Alt - Alt_x) + t_x \end{aligned} \tag{3.4}$$

Let the latitude and longitude of cell_0, cell_1,... , cell_n be (lat_0, lon_0), (lat_1, lon_1),... (lat_n, lon_n) and the tuples of cell_0, cell_1,... , cell_n at Alt be (d_0, f_0, t_0), (d_1, f_1, t_1),..., (d_n, f_n, t_n). At first the great circle distances from the point (Lat, Lon) to (lat_0, lon_0), (lat_1, lon_1),... (lat_n, lon_n) are calculated. These distances are called $Dist_0$, $Dist_1$, ... $Dist_n$. The 3 values of the tuple (d, f, t) at the point (Lat, Lon, Alt) are then calculated by Equation 3.5.

$$\begin{aligned} d &= \frac{\sum_{i=0}^{n} d_i\left(1 - \frac{Dist_i}{\sum_{j=0}^{n} Dist_j}\right)}{n} \\ f &= \frac{\sum_{i=0}^{n} f_i\left(1 - \frac{Dist_i}{\sum_{j=0}^{n} Dist_j}\right)}{n} \\ t &= \frac{\sum_{i=0}^{n} t_i\left(1 - \frac{Dist_i}{\sum_{j=0}^{n} Dist_j}\right)}{n} \end{aligned} \tag{3.5}$$

The number of neighboring cells (n) of cell_0 depends on the position of this cell. If the cell is in middle of the map, n is 8. If the cell is in the corner of the map, n is 3. If the cell is in edge of the map and is not in the corner, n is 5.

3.5 Simulation environment

In order to find the best 3-D UPR routes, algorithms in this thesis use a simulation environment to evaluate 3-D routes. The simulation environment using a 3-D route profile simulates the 4-D trajectory to evaluate the 3-D route.

The simulation of a 4-D trajectory for an aircraft, which is given aircraft performance, a 2-D (latitude, longitude) route, and altitude profile in a weather environment, includes the following three basic cases:

- The aircraft follows the 2-D route to climb from a starting point *(Lat1, Lon1, Alt1)* to the altitude $Alt2$ of an ending point. The segment from the starting to the ending point is called a climb segment. The outputs of this simulation include the latitude and longitude of the ending point, the weights (Distance, Time, Discomfort, Fuel), and the list of sub-segments of the segment.

- The aircraft follows the 2-D route to descend from a starting point *(Lat1, Lon1, Alt1)* to the altitude $Alt2$ of an ending point. The segment from the starting to the ending point is called a descent segment. The outputs of this simulation include the latitude and longitude *(Lat2, Lon2)* of the ending point, the weights, and the list of sub-segments of the segment.

- The aircraft cruise from a starting point *(Lat1, Lon1, Alt)* to an ending point *(Lat2, Lon2, Alt)*. The segment from the starting to the ending point is called a cruise segment. The outputs of this simulation include the weights, and the list of sub-segments of the segment.

Also, as the Top Of Descent (TOD) is not given, the simulation of the aircraft following the 2-D route to move backward from the destination to a given flight level is needed to determine the TOD at the flight level. In this thesis, we generalize this case as the simulation of the aircraft following the 2-D route to move backward from an ending point *(Lat2, Lon2, Alt2)* to the given altitude $Alt1$ of a starting point. The outputs of this simulation include the latitude and longitude *(Lat1, Lon1)* of the starting point, the weights, and the list of sub-segments of the segment from the starting point to the ending point.

In all cases the ending point of a segment is the starting point of the next segment, except for the ending point being the destination. The 3-D route of the aircraft is the series of these segments.

A segment following the 2-D route is simulated by the simulation of aircraft following the great circle route between every two waypoints of the 2-D route.

The mathematics of the evaluation environment relating to the atomic calculations of a moving step and the simulation of a whole route are presented in following sections.

3.6 The mathematics of the evaluation environment

The evaluation of an aircraft route is based on the simulation of a moving step. A number of variables need to be determined in a step. These include the change in altitude, the distance and time traveled, fuel burn, and the discomfort caused by bad weather cells. The calculation of these variables is based on the true air speed, the vertical speed (rate of climb or rate of descent), and the fuel flow of the aircraft. In this section, the calculation of true airspeed and vertical speed is presented first, followed by fuel flow calculation. Then the calculation of additional altitude, distance, time, fuel, and discomfort of a moving step is presented.

3.6.1 Calculation of True Airspeed and Vertical Speed

The problem is given aircraft type, flight phase, and the altitude (Alt) of the current position of an aircraft, with the condition that Alt is less than or equal to the maximum altitude the aircraft can reach. The task is determining true airspeed, and vertical speed.

From the aircraft type, the flight phase and BADA database, we can determine a set of true airspeeds and vertical speeds in different flight levels. Assume that the two closest flight levels from the set are Alt_x and Alt_{x+1}. The couples of true airspeeds and vertical speeds at these altitudes in the given flight phase are (TAS_x, VS_x) and (TAS_{x+1}, VS_{x+1}). The couple of (TAS, VS) at Alt is then linearly approximated by Equation 3.6.

$$
\begin{aligned}
TAS &= \frac{TAS_{x+1} - TAS_x}{Alt_{x+1} - Alt_x} \times (Alt - Alt_x) + TAS_x \\
VS &= \frac{VS_{x+1} - VS_x}{Alt_{x+1} - Alt_x} \times (Alt - Alt_x) + VS_x
\end{aligned} \tag{3.6}
$$

3.6.2 Calculation of Fuel Flow

The problem is given aircraft type, flight phase, altitude (*Alt*) and the temperature (*Temp*) of the current position, true airspeed (*TAS*), and weight (*W*) of an aircraft with the condition that *Alt* is less than or equal to the maximum altitude the aircraft can reach. The task is to calculate fuel flow for the aircraft at the current position.

The BADA aerodynamic model is used to calculate lift and drag as well as thrust and fuel flow in all flight phases [135]. It is used in air traffic simulation models of several organizations, research institutes and government organizations in ATM research such as Future ATM Concepts Evaluation Tool (FACET) of NASA [24], Early Demonstration & Evaluation Platform (eDEP) of Eurocontrol [52], and the Air Traffic Operations and Management Simulator (ATOMS) [3]. Most aircraft types currently modeled in BADA demonstrate mean error in vertical speed lower than 100 feet per minute (fpm), over previously specified normal operation conditions. This corresponds to mean error of vertical speed lower than 5% for operational range of flight envelope under the assumption that 2000 fpm is an average vertical speed value. Fuel consumption is modeled with mean error lower than 5% for the same conditions [136].

The aircraft type is used to determine aircraft performance parameters from BADA database. This database is developed by the Eurocontrol Experimental Centre [135]. Version 3.6 includes detailed information on 91 supported aircraft types, gathered from reference sources like flight and operating manuals. BADA includes a total of 51 parameters per aircraft, which specify the aircraft's mass and flight envelope together with its aerodynamics, engine thrust, and fuel flow coefficients.

The thrust T is calculated first. Then fuel flow is calculated based on the thrust.

The thrust, T (N), in general is calculated as follows with the input of true air speed (TAS), altitude, temperature, and weight (W) and operations performance parameters from BADA database. The details of calculation is presented in BADA document [135].

- If the flight is in cruise phase, thrust is the minimum of the maximum cruise thrust and drag. The maximum cruise thrust and drag are in Sections 3.7.2 and 3.6.1 in BADA document [135] respectively.
- If the flight is in climb or take off phases, the thrust is the maximum climb or take off thrust. Their calculations are presented in Section 3.7.1 in BADA document [135].
- If the flight is in descent phase, the thrust is the descent thrust. Its calculation is presented in Section 3.7.3 in BADA document [135].

The thrust specific fuel consumption, η, in kg/minute/kN is specified as a function of true airspeed, TAS (knots), for the jet and turboprop engines. The nominal fuel flow, f_{nom} (kg/minute), can then be calculated using the thrust, T. These auscultations also use two BADA performance paraments (C_{f1} and C_{f2}): the first and second thrust specific fuel consumption coefficients.

$$jet: \quad \eta = C_{f1} \times (1 + \frac{TAS}{C_{f2}}) \tag{3.7}$$

where:

$$f_{nom} = \eta \times T \tag{3.8}$$

$$turboprop: \quad \eta = C_{f1} \times (1 - \frac{TAS}{C_{f2}}) \times (TAS/1000) \tag{3.9}$$

where:

$$f_{nom} = \eta \times T \tag{3.10}$$

These expressions are used in all flight phases except during cruise and for descent/idle conditions.

Minimum fuel flow, f_{min} , corresponding to idle thrust or descent conditions for both jet and turboprop engines, is specified in kg/minute as a function of altitude above sea level, *Alt* (ft). This calculation uses two BADA performance paraments (C_{f3} and C_{f4}): the first and second descent fuel flow coefficients.

$$jet/turboprop: \quad f_{min} = C_{f3}(1 - \frac{Alt}{C_{f4}}) \tag{3.11}$$

Cruise fuel flow, f_{cr}, is calculated using the thrust specific fuel consumption η and a BADA performance parameter (C_{fcr}): cruise fuel flow factor.

$$jet/turboprop: \quad f_{cr} = \eta \times T \times C_{fcr} \tag{3.12}$$

For piston engines the fuel flow, *f*, in kg/minutes is specified to be a constant, that is,

$$f_{cr} = C_{f1} \times C_{f2} \quad (cruise) \tag{3.13}$$

$$f_{min} = C_{f3} \quad (idle/descent) \tag{3.14}$$

$$f_{nom} = C_{f1} \quad (all\,other\,phases) \tag{3.15}$$

3.6.3 Additional Altitude, Distance, Time, Fuel, and Discomfort of a Moving Step

The problem is given the current point of an aircraft A, the horizontal target point C, the magnitude of aircraft true airspeed $\overrightarrow{TAS}$, the vector of aircraft vertical speed $\overrightarrow{V_y}$, and the vector of wind speed $\overrightarrow{WS}$ as in Figure 3.7; also the bad weather level (BWL), and the fuel flow (FF) of the aircraft at A. The task is determining additional altitude h, distance traveled d, the time duration t, discomfort dc, and fuel burn f when the aircraft moves from A to F.

From $|\overrightarrow{TAS}|$ and $|\overrightarrow{V_y}|$, we can calculate the magnitude of horizontal aircraft

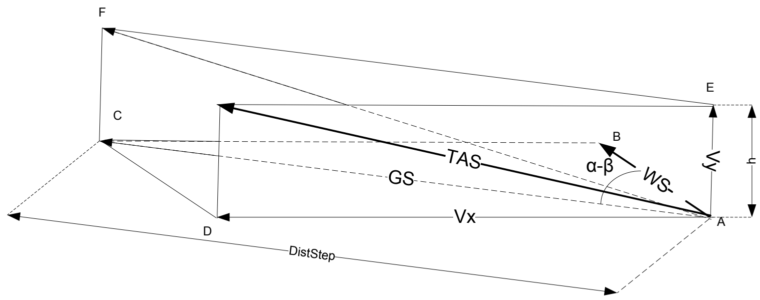


Figure 3.7: A Climb Moving Step.

speed $|\overrightarrow{V_x}|$ as Equation 3.16.

$$V_x = \sqrt{TAS^2 - V_y^2} \tag{3.16}$$

We assume that AC is short enough that while the flight is moving from A to F, $\overrightarrow{WS}$, $\overrightarrow{V_x}$, $\overrightarrow{V_y}$, and FF are the same as at A.

To reach the horizontal target C, the flight needs to have a proper heading, so that when the flight movement is affected by $\overrightarrow{WS}$, it can horizontally reach the destination. In Figure 3.16, the flight chooses the heading $\overrightarrow{AD}$ to go to C, as $\overrightarrow{AC} = \overrightarrow{AD} + \overrightarrow{AB}$, where $\overrightarrow{AD}$ presents the movement of the flight by $\overrightarrow{V_x}$ and $\overrightarrow{AB}$ presents the movement of the flight by $\overrightarrow{WS}$.

As the two points A, C, and $\overrightarrow{WS}$ are known, the initial bearing from A to C and the wind direction are known. We call them α and β respectively. These angles are measured in degrees. Then the angle $\widehat{BAC}$ is determined as $\alpha - \beta$. According to the law of cosines in a triangle, we have Equation 3.17.

$$cos(\alpha - \beta) = \frac{AC^2 + AB^2 - BC^2}{2AC * AB} \tag{3.17}$$

$$\Rightarrow cos(\alpha - \beta) = \frac{AC^2 + (WS * t)^2 - (V_x * t)^2}{2AC * WS * t} \tag{3.18}$$

$$\Rightarrow (V_x^2 - WS^2) * t^2 + 2AC * WS * cos(\alpha - \beta) * t - AC^2 = 0 \tag{3.19}$$

We can assume that $V_x > WS$ and $AC > 0$, so the quadratic equation 3.19 with variable t has two roots. One root is greater than 0, and the other is less than 0.

The root which is greater than 0 is t.

$$t = \frac{AC(\sqrt{V_x^2 - WS^2 sin(\alpha - \beta)^2} - WScos(\alpha - \beta))}{V_x^2 - WS^2} \tag{3.20}$$

If ABC is not a triangle, $\alpha - \beta$ will be $0°$ or $180°$. We can also consider the case when WS is 0 as when $\alpha - \beta$ is 0 and use the same set of following equations.

If $\alpha - \beta$ is $0°$, the ground speed is calculated by Equation 3.21

$$GS = V_x + WS \tag{3.21}$$

If $\alpha - \beta$ is 180, the ground speed is calculated by Equation 3.22

$$GS = V_x - WS \tag{3.22}$$

Then t is calculated by Equation 3.23

$$t = \frac{AC}{GS} \tag{3.23}$$

After having time duration t, we calculate the additional altitude h by Equation 3.24

$$h = V_y * t \tag{3.24}$$

Then the distance traveled is calculated by Equation 3.25

$$d = \sqrt{AC^2 + h^2} \tag{3.25}$$

The fuel consumption f is calculated by Equation 3.26

$$f = FF * t \tag{3.26}$$

These equations are applied to all types of aircraft movement: climb, cruise and descend. If the aircraft climbs, $V_y > 0$. If the aircraft cruises, $V_y = 0$. If the aircraft descends, $V_y < 0$.

The discomfort of a moving step is calculated by Equation 3.27, where BWL is the bad weather level at the first point of the moving step.

$$dc = t * BWL \tag{3.27}$$

3.7 4-D trajectory simulation

In this section, the simulations of all the basic segments in simulating a route description to generate a 4-D trajectory with high resolution are presented.

3.7.1 Simulating a 2-D route to climb to an altitude

The problem is given the weather environment, aircraft performance, a 2-D (latitude, longitude) route, a starting point $(Lat1, Lon1, Alt1)$ and an altitude $Alt2$ $(Alt2 > Alt1)$. The task is to simulate aircraft climbing from the starting point to the altitude $Alt2$, following the 2-D route. The output includes latitude, longitude, and altitude of the ending point $(Lat2, Lon2, Alt2)$ (where $Alt2$ is already given), the weights, and the list of sub-segments of the climb segment.

Algorithm 1 is to resolve the problem. The simulation of the climb segment begins with the starting point. This point is considered as the current simulated point *currentpoint* of the aircraft. The next waypoint *nextpoint* to the current point in the route is determined. If *nextpoint* is not available, the simulation terminates. Otherwise the destination altitude Alt, the weights and the list of sub-segments when the aircraft climbs from *currentpoint* to *nextpoint* following the great circle route between the two points are determined. This will be presented in Section 3.7.2 ($\star$). If Alt is higher than $Alt2$ or Alt is not determined, a point at altitude $Alt2$ between *currentpoint* and *nextpoint*, the weights, and the list of sub-segments when the aircraft follows the great circle route from *currentpoint* to *nextpoint* to climb to $Alt2$ are determined. This will be presented in Section 3.7.3 ($*$). The weights of the segment and the list of sub-segments are updated by those determined in ($*$) and the simulation finishes. If *Alt* is equal to *Alt*2, the weights of the segment and the list of sub-segments are updated by those determined in ($\star$) and the simulation finishes. If *Alt* is lower than *Alt*2, a new point *newpoint* is created with latitude and longitude being those of the *nextpoint* and altitude of *Alt*. The weights of the segment and the list of sub-segments are updated by those determined in ($\star$). If there is a point next to *nextpoint* in the route, *currentpoint* becomes *newpoint* and *nextpoint* becomes the point next to *nextpoint*. Otherwise the simulation finishes.

Algorithm 1 The algorithm for the simulation of aircraft following a 2-D route to climb from a starting point to a point given altitude.

```
currentpoint = startingpoint.
nextpoint = the next point of currentpoint in the 2-D route.
if nextpoint is available then
    Finish = false
    while not Finish do
        Determine the altitude Alt, the weights, and the list of sub-segments when aircraft climbs from the
        currentpoint to the nextpoint. This will be presented in Section 3.7.3 (⋆).
        if Alt > Alt2 or Alt is NaN then
            Determine a point between currentpoint and nextpoint, the weights, and the list of sub-segments
            when aircraft climbs from currentpoint to the point when the flight follows the great circle route from
            currentpoint to nextpoint to climbs to Alt2. This will be presented in Section 3.7.2 (∗).
            The weights of the segment and the list of subsegments are updated by those determined in (∗).
            Finish = true
        else
            if Alt = Alt2 then
                The weights of the segment and the list of subsegments are updated by those determined in (⋆).
                Finish = true
            else
                Create a new point newpoint whose latitude and longitude are those of the nextpoint and altitude
                is Alt.
                The weights of the segment and the list of subsegments are updated by those determined in (⋆).
                if there is a point next to nextpoint then
                    currentpoint = newpoint
                    nextpopint = the point next to nextpoint.
                else
                    Finish = true
                end if
            end if
        end if
    end while
end if
```

Figure 3.8 is an example of a descent segment. The aircraft follows 2-D (latitude, longitude) route of 4 consecutive waypoints j, $j+1$, $j+2$, and $j+3$ to climb from the starting point $(Lat1, Lon1, Alt1)$ to the altitude $Alt2$. The figure shows points of the climb segment at the places the aircraft passes waypoints of the 2-D route, the starting point, and the ending point by symbol ▲.

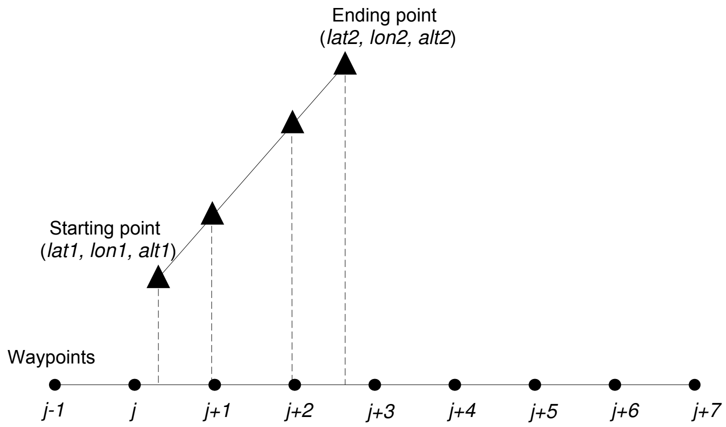


Figure 3.8: An example of the simulation of aircraft following a 2-D route to climb from a starting point to the given altitude of an ending point.

3.7.2 Simulating a great circle route to find the closest climb altitude

The problem is given the weather environment, aircraft performance, a starting point S $(Lat1, Lon1, Alt1)$, and the latitude and longitude $(Lat2, Lon2)$ of an ending point E. The task is to simulate the aircraft following the great circle route to climb from S to E. The outputs include the altitude of the ending point $Alt2$ $(Alt2 > Alt1)$, the weights, and the list of sub-segments of the climb segment. If the aircraft climbs to the maximum cruising altitude but it still does not pass the ending point, $Alt2$ is not determined and it is given a value of NaN (Not a Number). In computing, NaN is a value of the numeric data type representing an undefined or unrepresentable value.

Figure 3.9 presents the algorithm flow for the simulation of aircraft climbing from a starting point S *(Lat1, Lon1, Alt1)* to the latitude and longitude *(Lat2, Lon2)* of an ending point E. The simulation is implemented by flying the aircraft step by step following the great circle route from S to E. At each step the aircraft moves a horizontal distance of $DistStep$, except in the final step the aircraft may move directly to E. The simulation finishes when the aircraft moves to E or the aircraft climbs to the maximum cruising altitude but it does not reach E.

Initially the weights of the segment are 0, and the list of sub-segments is empty.

The simulation begins with the starting point, which is considered as the current simulated point *currentpoint* of the aircraft. If the horizontal distance from the current point to the ending point is less than $DistStep$, latitude and longitude ($Lat3$ and $Lon3$) of the next point (which is horizontally $DistStep$ by distance next to the current point in the great circle route from S to E) are determined. Otherwise $Lat3$ and $Lon3$ are the latitude and longitude of the ending point E ($Lat2$ and $Lon2$).

After determining the latitude and longitude of the next point of the aircraft, following steps are implemented.

- Bad weather level at the current point is determined as that of the weather cell containing it. The weather cell containing the current point is found in the bad weather grid (see Section 3.3).
- Determine wind direction, wind speed, and temperature at the current point (as in Section 3.4.2).
- Determine climb true airspeed TAS, vertical speed Vy of the aircraft at the current point (as in Section 3.6.1).
- Determine fuel flow at the current point (as in Section 3.6.2).
- Determine additional altitude of the next point, distance travelled, time, discomfort, and fuel burn while the aircraft moves from the current point to the next point (as in Section 3.6.3).
- Altitude of the next point $Alt3$ is the altitude of the current point plus additional altitude.
- Insert the sub-segment from the current point to the next point into the sub-segment list of the segment.
- Increase the weights of the segment by the weights of the sub-segment.

When the simulation finishes, the altitude ($Alt2$) of the ending point E is that of the current point.

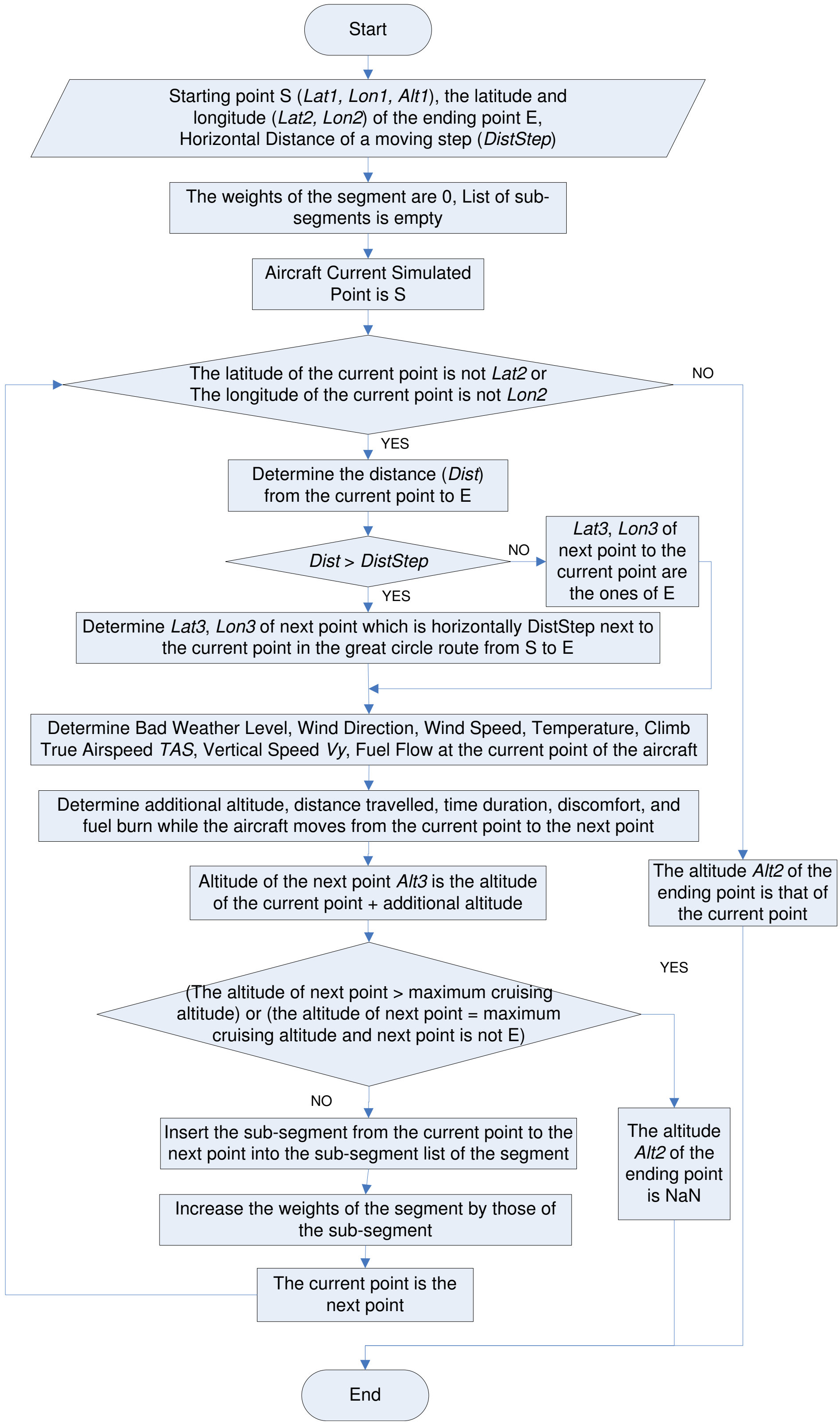


Figure 3.9: Algorithm flow for the simulation of aircraft following a great circle route to climb from a starting point to a point given latitude, longitude

3.7.3 Simulating a great circle route to climb to a target altitude

The problem is given the weather environment, aircraft performance, a starting point S ($Lat1, Lon1, Alt1$), and the altitude $Alt2$ of an ending point E ($Alt2 > Alt1$), and an oriented point O (Lat, Lon) which is the waypoint next to the starting point S. O is far enough that aircraft will not pass O when it reaches $Alt2$. The task is to simulate aircraft following the great circle route from S to O to climb from S to the altitude $Alt2$ of the ending point E. The output includes the latitude, and longitude of the ending point ($Lat2, Lon2$), the weights, and the list of sub-segments of the climb segment.

Figure 3.10 presents the algorithm flow for the simulation of aircraft following the great circle route from the starting point S ($Lat1, Lon1, Alt1$) to the oriented point O to climb from the starting point to the altitude $Alt2$ of the ending point. The simulation is implemented by flying the aircraft step by step following the great circle route from S to O. At each step the aircraft moves a horizontal distance of $DistStep$, until it reaches altitude $Alt2$.

Initially the weights of the segment are 0, and the list of sub-segments is empty. The simulation begins with the starting point, which is considered as the current simulated point *currentpoint* of the aircraft. If the difference between the altitude of the current point and $Alt2$ is greater than a user-defined value y, the latitude and longitude ($Lat3$ and $Lon3$) of the next point (which is horizontally $DistStep$ by distance next to the current point in the great circle route from S to E) are determined. Otherwise the simulation finishes.

The altitude of the next point $Alt3$, the weights of the sub-segment from the current point to the next point is calculated by steps as presented in Section 3.7.2. Then the sub-segment from the current point to the next point is inserted into the sub-segment list of the segment. The weights of the segment increase by those of the sub-segment.

When the simulation finishes, the latitude ($Lat2$) and longitude ($Lon2$) of the

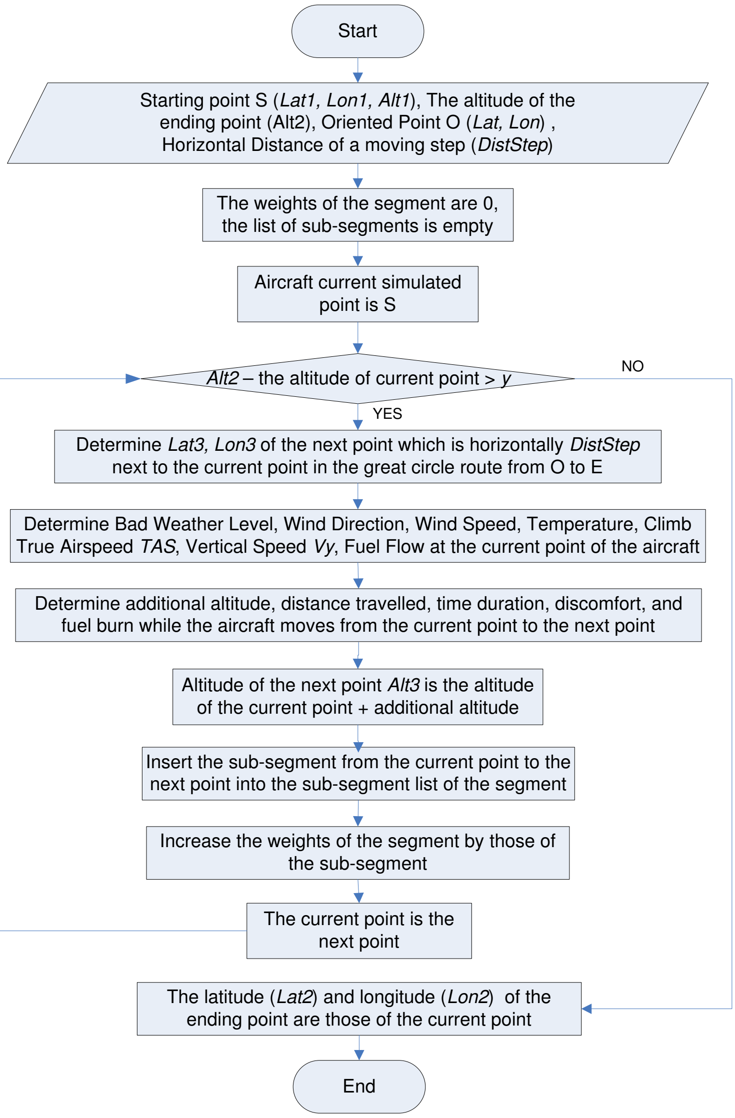


Figure 3.10: Algorithm flow for the simulation of an aircraft following a great circle route to climb from a starting point to a point given altitude

ending point E are those of the current point.

3.7.4 Simulating a 2-D route to descend from a starting point to a target altitude

The problem is given the weather environment, aircraft performance, a 2-D (latitude, longitude) route, a starting point $(Lat1, Lon1, Alt1)$ and the altitude $Alt2$ of an ending point $(Alt2 < Alt1)$. The task is to simulate aircraft descending from the starting point to the altitude $Alt2$, following the 2-D route. The output includes the latitude, longitude, and altitude of the ending point $(Lat2, Lon2, Alt2)$ (where $Alt2$ is already given), the weights, and the list of sub-segments of the descent segment.

Algorithm 2 is to resolve the problem. The simulation of the descent segment begins with the starting point. This point is considered as the current simulated point *currentpoint* of the aircraft. The next waypoint *nextpoint* to the current point in the route is determined. If *nextpoint* is not available, the simulation finishes. Otherwise the destination altitude Alt, the weights, and the list of sub-segments when the aircraft descends from *currentpoint* to *nextpoint* following the great circle route between the two points are determined. This will be presented in Section 3.7.5 ($\star$). If Alt is lower than $Alt2$, a point at altitude $Alt2$ between *currentpoint* and *nextpoint*, the weights, and the list of sub-segments when the aircraft follows the great circle route from *currentpoint* to *nextpoint* to descend to $Alt2$ are determined. This will be presented in Section 3.7.6 ($*$). The weights of the segment and the list of sub-segments are updated by those determined in ($*$) and the simulation finishes. If Alt is equal to $Alt2$, the weights of the segment and the list of sub-segments are updated by those determined in ($\star$) and the simulation finishes. If Alt is lower than $Alt2$, a new point *newpoint* is created with the latitude and longitude being those of the *nextpoint* and altitude of Alt. The weights of the segment, and the list of sub-segments are updated by those determined in ($\star$). If there is a point next to *nextpoint* in the route, *currentpoint* becomes *newpoint* and *nextpoint* becomes the point next to *nextpoint*. Otherwise the simulation finishes.

Figure 3.11 is an example of a climb segment. The aircraft follows 2-D (latitude, longitude) route of 5 consecutive waypoints j, $j+1$, $j+2$, $j+3$, and $j+4$ to climb from the starting point $(Lat1, Lon1, Alt1)$ to the altitude $Alt2$ of the ending point. The figure shows points of the descent segment at the places the aircraft passes waypoints of the 2-D route, the starting point, and the ending point by symbol $\blacktriangledown$.

Algorithm 2 The algorithm for the simulation of aircraft following a 2-D route to descend from a starting point to a point given altitude

```
currentpoint = the starting point (Lat1, Lon1, Alt1).
nextpoint = the point next to currentpoint in the route.
if nextpoint is available then
    Finish = false
    while not Finish do
        Determine the altitude Alt, the weights, and the list of sub-segments when the aircraft descends from the
        currentpoint to the nextpoint. This will be presented in Section 3.7.5. (⋆)
        if Alt < Alt2 then
            Determine a point between currentpoint and nextpoint, the weights, and the list of sub-segments
            when the aircraft follows the great circle route from currentpoint to nextpoint to descend to Alt2.
            This will be presented in Section 3.7.6. (∗)
            The weights of the segment and the list of sub-segments are updated by those determined in (∗).
            Finish = true
        else
            if The altitude = FL_{i-1} then
                The weights of the segment and the list of sub-segments are updated by those determined in (⋆).
                Finish = true
            else
                Create a new point newpoint whose latitude and longitude are those of the nextpoint and its
                altitude is Alt.
                The weights of the segment and the list of sub-segments are updated by those determined in (⋆).
                if there is a point next to nextpoint in the route then
                    currentpoint = newpoint
                    nextpoint = the point next to nextpoint in the route.
                else
                    Finish = true
                end if
            end if
        end if
    end while
end if
```

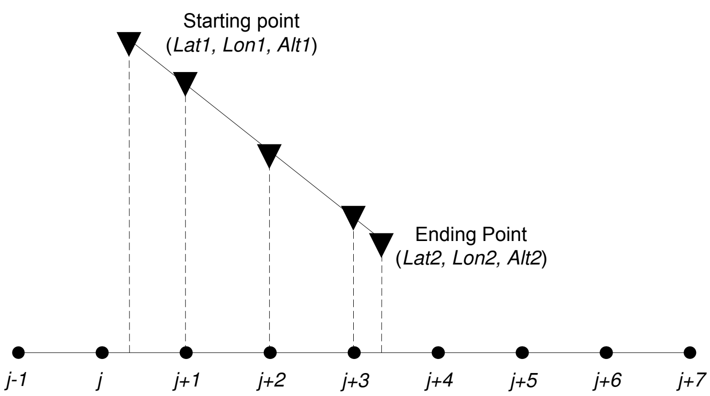


Figure 3.11: An example of the simulation of aircraft following a 2-D route to descend from a starting point to a point given altitude

3.7.5 Simulating a great circle route to descend from a starting point to a target latitude and longitude

The problem is given the weather environment, aircraft performance, a starting point S *(Lat1, Lon1, Alt1)*, and the latitude and longitude *(Lat2, Lon2)* of an ending point E. The task is to simulate aircraft following the great circle route to descend from S to E. The outputs include the altitude of the ending point $Alt2$ ($Alt2 < Alt1$), the weights, and the list of sub-segments of the descent segment.

Figure 3.12 presents the algorithm flow for the simulation of aircraft descending from the starting point S ($Lat1, Lon1, Alt1$) to the latitude and longitude ($Lat2, Lon2$) of the ending point E. The simulation is implemented by flying the aircraft step by step following the great circle route from S to E. At each step the aircraft moves a horizontal distance of *DistStep*, except in the final step the aircraft may move directly to E. The simulation finishes when the aircraft moves to E or the aircraft descends to the altitude of 3000 feet but it does not reach E. Here the aircraft only descends to a point which is not the destination, so the constraint that altitude is higher than or equal to 3000 feet of the ending point is used. The aircraft descending from TOD to the destination is simulated by backward simulation in Section 3.7.7.

Initially the weights of the segment are 0, and the list of sub-segments is empty. The simulation begins with the starting point, which is considered as the current simulated point *currentpoint* of the aircraft. If the horizontal distance from the current point to the ending point is less than *DistStep*, the latitude and longitude ($Lat3$ and $Lon3$) of the next point (which is horizontally *DistStep* by distance next to the current point in the great circle route from S to E) are determined. Otherwise $Lat3$ and $Lon3$ are the latitude and longitude of the ending point E ($Lat2$ and $Lon2$).

The altitude of the next point $Alt3$, and the weights of the sub-segment from the current point to the next point are calculated as in Section 3.7.2. Then the sub-segment from the current point to the next point is inserted into the sub-segment list of the segment. The weights of the segment increase by those of the sub-segment.

When the simulation finishes, the altitude ($Alt2$) of the ending point E is that of the current point.

3.7.6 Simulating a great circle route to descend from a starting point to a target altitude

The problem is given the weather environment, aircraft performance, the starting point S ($Lat1, Lon1, Alt1$), and the altitude $Alt2$ of the ending point E ($Alt2 < Alt1$), and an oriented point O (Lat, Lon) which is the waypoint next to the starting point S. O is far enough that aircraft will not pass O when it reaches $Alt2$. The task is to simulate aircraft following the great circle route from S to O to descend from S to the altitude $Alt2$ of the ending point E. The output includes the latitude, and the longitude of the ending point ($Lat2, Lon2$), the weights, and the list of sub-segments of the descent segment.

Figure 3.13 presents the algorithm flow for the simulation of aircraft following the great circle route from the starting point S ($Lat1, Lon1, Alt1$) to the oriented point O to descend from the starting point to the altitude $Alt2$ of the ending point. The simulation is implemented by flying the aircraft step by step following the great circle route from S to O. At each step the aircraft moves a horizontal distance of $DistStep$, until it reaches altitude $Alt2$.

The weights of the segment are firstly initialized as 0. The simulation begins with the starting point, which is considered as the current simulated point $currentpoint$ of the aircraft. If the difference between the altitude of the current point and $Alt2$ is greater than a user-defined value y, the latitude and longitude ($Lat3$ and $Lon3$) of the next point (which is horizontally $DistStep$ by distance next to the current point in the great circle route from S to E) are determined. Otherwise the simulation finishes.

The altitude of the next point $Alt3$, and the weights of the sub-segment from the current point to the next point are calculated as in Section 3.7.2. Then the subsegment from the current point to the next point is inserted into the sub-segment list of the segment. The weights of the segment increase by those of the sub-segment.

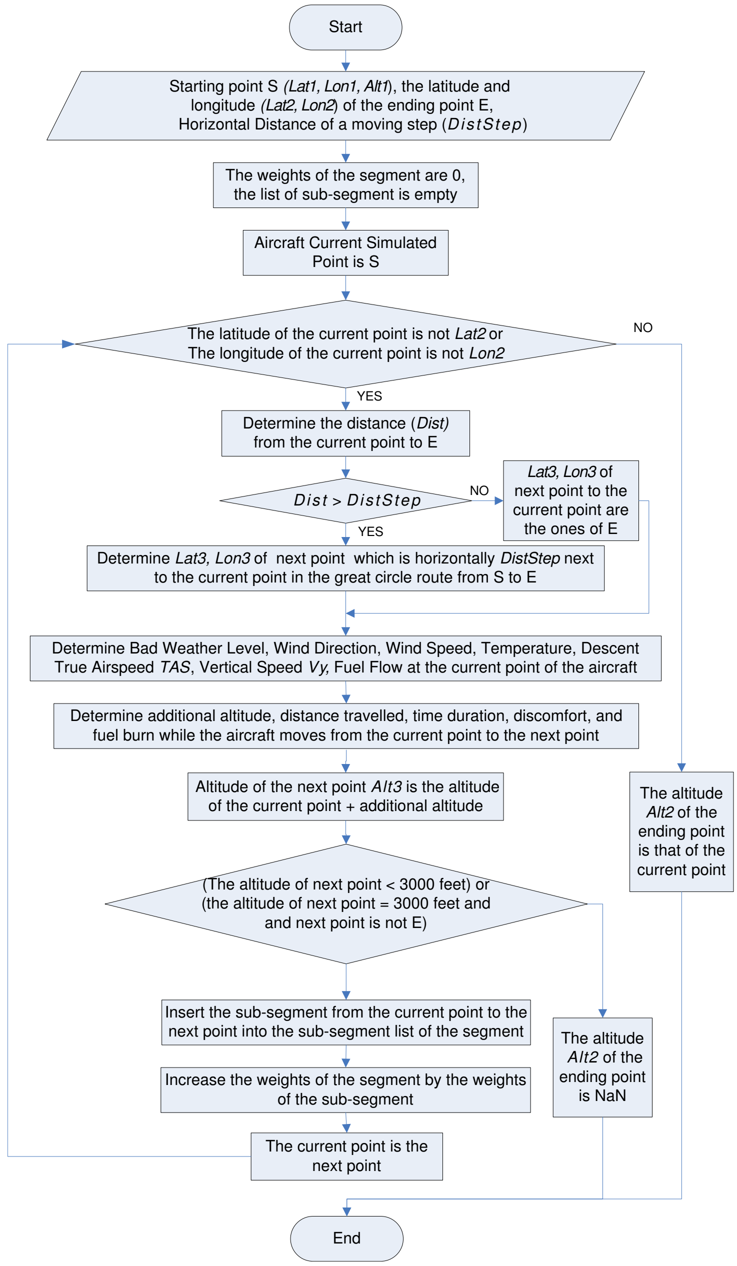


Figure 3.12: Algorithm flow for the simulation of an aircraft following the great circle route to descend from a point given latitude, longitude and altitude to another point given latitude and longitude

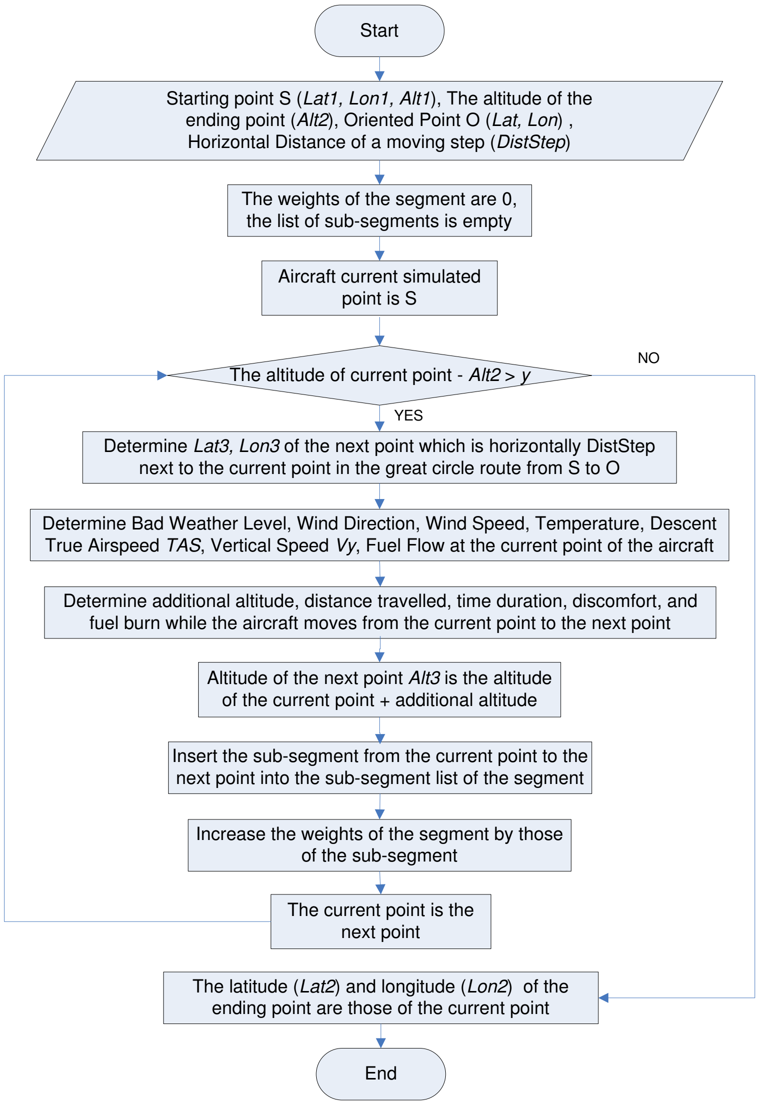


Figure 3.13: Algorithm flow for the simulation of aircraft following a great circle route to descend from a starting point to a point given altitude

When the simulation finishes, the latitude ($Lat2$) and longitude ($Lon2$) of the ending point E are those of the current point.

3.7.7 Backward simulation of aircraft following a 2-D route to descend from a starting point given altitude to an ending point

The problem is given the weather environment, aircraft performance, a 2-D (latitude, longitude) route, the altitude $Alt1$ of a starting point, and an ending point $(Lat2, Lon2, Alt2)$ $(Alt2 < Alt1)$. The task is to determine the latitude, longitude of the starting point $(Lat1, Lon1)$ (the altitude of the starting point $Alt1$ is already given) so that the aircraft can follow the 2-D route to descend from this point to the ending point, and also to determine the weights, and the list of sub-segments of the descent segment.

Backward simulation is used to determine these variables. The simulation begins with the ending point. This point is considered as the current simulated point *currentpoint* of the aircraft. The previous waypoint *prepoint* of the current point in the route is determined. If *prepoint* is not available, the simulation finishes. Otherwise the altitude Alt, the weights, and the list of sub-segments are determined by simulating aircraft's movement backward from *currentpoint* to *prepoint* with aircraft performance in descent phase. This will be presented in Section 3.7.8 ($\star$). If Alt is higher than $Alt1$, or Alt is not determined, the destination point (which is between *currentpoint* and *prepoint*), the weights, and the list of sub-segments when the aircraft follows the great circle route from *currentpoint* to *prepoint* to move backward to $Alt1$ are determined. This will be presented in Section 3.7.9 ($*$). Then the weights of the segment and the list of sub-segments are updated by those determined in ($*$) and the simulation finishes. If Alt is equal to $Alt1$, the weights of the segment, and the list of sub-segments are updated by those determined in ($\star$), and the simulation finishes. If Alt is lower than $Alt1$, a new point *newpoint* whose latitude and longitude are those of the *prepoint*, and its altitude is Alt is created. The weights of the segment and the list of sub-segments are updated by those determined in ($\star$). If there is a point previous to *prepoint* in the route, *currentpoint* becomes *prepoint* and *prepoint* becomes the point previous to *prepoint*. Otherwise the simulation finishes.

The details of the algorithm for backward simulation of aircraft following the 2-D route to descend from a starting point given altitude to an ending point is in Algorithm 3. Figure 3.14 is an example of a descent segment in this case. The

Algorithm 3 Backward simulation of aircraft following a 2-D route to descend from a starting point given altitude to an ending point

```
currentpoint = the ending point E.
prepoint = the previous point of currentpoint in the route.
if prepoint is available then
    Finish = false
    while not Finish do
        Determine the altitude Alt, the weights, and the list of sub-segments by simulating aircraft's movement
        backward from currentpoint to prepoint with aircraft performance in descent phase. This will be pre-
        sented in Section 3.7.8 (⋆).
        if Alt > Alt1 then
            Determine the destination point between currentpoint and prepoint, the weights, and the list of
            sub-segments when the aircraft follows the great circle route from currentpoint to prepoint to move
            backward to Alt1. This will be presented in Section 3.7.9 (∗).
            The weights of the segment and the list of sub-segments are updated by those determined as in (∗).
            Finish = true.
        else
            if Alt = Alt1 then
                The weights of the segment and the list of sub-segments are updated by those determined as in
                (⋆).
                Finish = true.
            else
                Create a new point newpoint whose latitude and longitude are those of the prepoint and its altitude
                is Alt.
                The weights of the segment and the list of sub-segments are updated by those determined as in
                (⋆).
                if there is a point previous to prepoint in the route then
                    currentpoint = newpoint
                    prepoint = the point previous to prepoint in the route.
                else
                    Finish = true
                end if
            end if
        end if
    end while
end if
```

aircraft follows the 2-D (latitude, longitude) route of 5 consecutive waypoints j, $j+1$, $j+2$, $j+3$ and $j+4$ to move backward from the ending point to the altitude $Alt1$ of the starting point. The points where the aircraft passes waypoints of the 2-D route, the starting point, and the ending point are presented by symbol ▼.

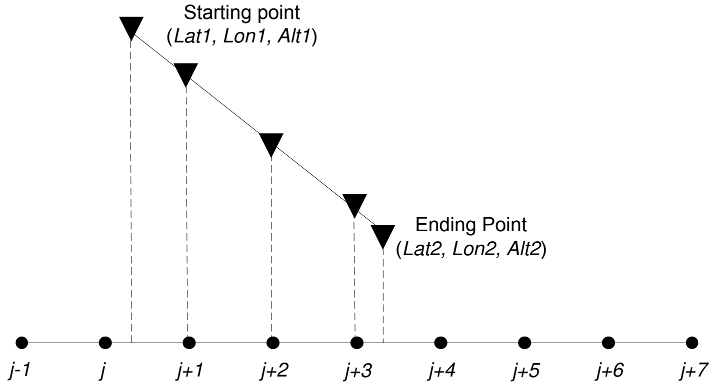


Figure 3.14: Backward simulation of aircraft following a 2-D route to descend from a starting point given altitude to an ending point

3.7.8 Backward Simulation of aircraft following the great circle route to descend from a starting point given latitude and longitude to an ending point

The problem is given the weather environment, aircraft performance, a 2-D (latitude, longitude) route, the latitude and longitude $(Lat1, Lon1)$ of a starting point S, and an ending point $(Lat2, Lon2, Alt2)$ E. The task is to determine the altitude of the starting point $Alt1$ ($Alt1 > Alt2$) so that the aircraft can follow the great circle route to descend from the starting point to the ending point, and also to determine the weights, and the list of sub-segments of the descent segment.

Figure 3.15 presents the algorithm flow for the simulation of aircraft moving backward from the ending point E ($Lat2$, $Lon2$, $Alt2$) to the latitude and longitude ($Lat1$, $Lon1$) of the starting point S. The backward simulation is implemented by flying the aircraft backward step by step following the great circle route from S to E. At each step the aircraft moves a horizontal distance of $DistStep$, except the final step which may move backward directly to S. The simulation finishes when the aircraft moves to S or the aircraft reaches the maximum cruising altitude but it does not reach S. If the aircraft reaches the maximum cruising altitude but it still does not reach the ending point, $Alt2$ is not determined and it is given a value of NaN.

Initially the weights of the segment are 0 and the list of sub-segments is empty.

The simulation begins with the ending point, which is considered as the current simulated point *currentpoint* of the aircraft. If the horizontal distance from the current point to the starting point is less than $DistStep$, the latitude and longitude ($Lat3$, $Lon3$) of the previous point which is horizontally $DistStep$ by distance previous to the current point in the great circle route from S to E are determined. Otherwise $Lat3$ and $Lon3$ are the latitude and longitude ($Lat1$, $Lon1$) of the starting point S.

The altitude of the next point $Alt3$, the weights of the sub-segment from the current point to the previous point is calculated by steps as presented in Section 3.7.2. Then the sub-segment from the current point to the next point is inserted into the sub-segment list of the segment. The weights of the segment increase by those of the sub-segment.

When the simulation finishes, the altitude ($Alt1$) of the starting point S is that of the current point.

3.7.9 Backward simulation of aircraft following the great circle route to descend from a starting point given altitude to an ending point

The problem is given the weather environment, aircraft performance, a 2-D (latitude, longitude) route, the altitude $Alt1$ of the starting point, the ending point ($Lat2, Lon2, Alt2$), an oriented point O (Lat, Lon) which is the waypoint previous to the ending point in the route. O is far enough that aircraft will not pass O when it reaches $Alt1$. The task is to determine the latitude, and longitude ($Lat1, Lon1$) of the starting point so that the aircraft can follow the great circle route from O to E to descend from this point to the ending point. The task is also to determine the weights, and the list of sub-segments of the descent segment.

Figure 3.16 presents the simulation of aircraft moving backward from an ending point E ($Lat2, Lon2, Alt2$) to the altitude $Alt1$ of the starting point S. The backward simulation is implemented by flying the aircraft backward step by step following the

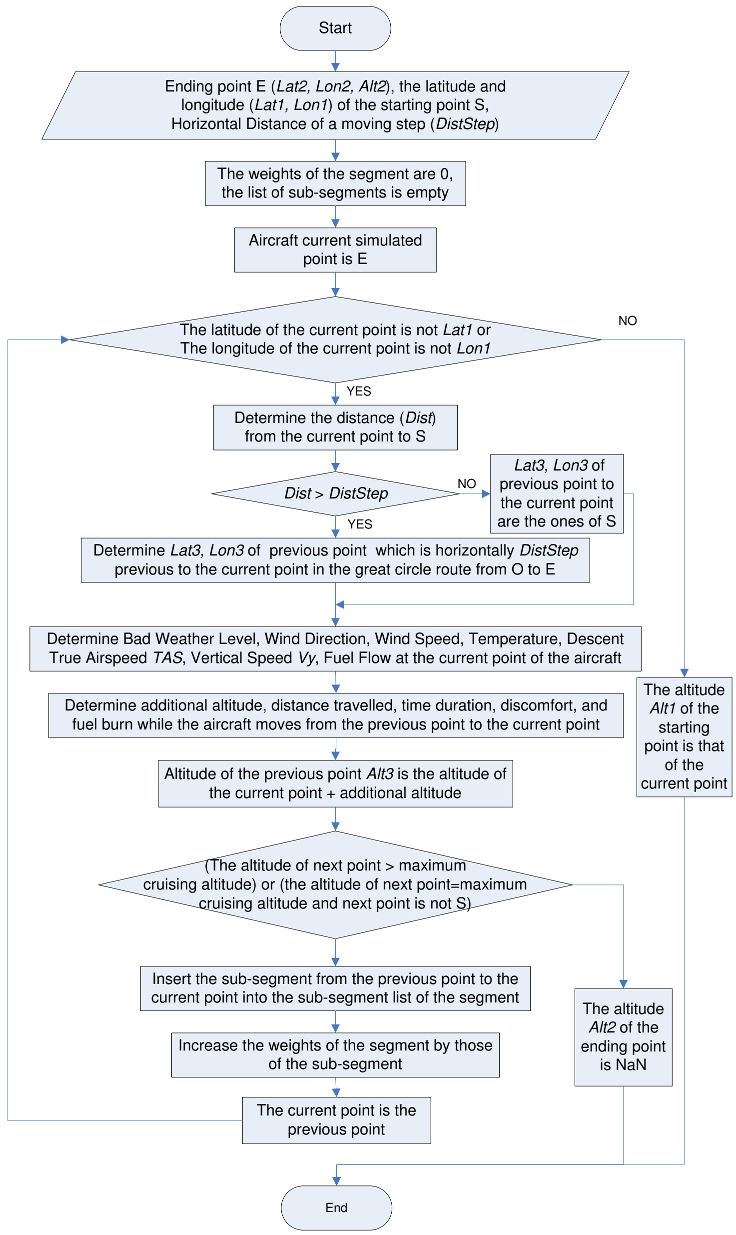


Figure 3.15: Algorithm flow for the backward simulation of aircraft following the great circle route to descend from a starting point given latitude and longitude to an ending point

great circle route from S to E until it reaches $Alt1$.

Initially the weights of the segment are 0, and the list of sub-segments is empty. The simulation begins with the ending point, which is considered as the current simulated point *currentpoint* of the aircraft. If the difference between $Alt1$ and the altitude of the current point is greater than a user-defined value y, latitude and longitude ($Lat3$, $Lon3$) of the previous point (which is horizontally *DistStep* by distance previous to the current point in the great circle route from O to E) are determined. Otherwise the simulation finishes.

The altitude of the previous point $Alt3$, and the weights of the sub-segment from the previous point to the current point are calculated as in Section 3.7.2. Then the sub-segment from the previous point to the current point is inserted into the sub-segment list of the segment. The weights of the segment increase by those of the sub-segment.

When the simulation finishes, the latitude ($Lat1$) and longitude ($Lon1$) of the starting point are those of the current point.

3.7.10 Simulation of aircraft following a 2-D route to cruise from a starting point to an ending point

The problem is given the weather environment, aircraft performance, a 2-D (latitude, longitude) route, the starting point S ($Lat1, Lon1, Alt$), and the ending point E ($Lat2, Lon2, Alt$). The task is to determine the list of sub-segments, and the weights when the aircraft follows the 2-D route to cruise from S to E.

The simulation is implemented by flying the aircraft step by step following the 2-D route from S to E. The simulation begins with the starting point, which is considered as the current simulated point *currentpoint* of the aircraft. If the horizontal distance from *currentpoint* to the ending point or the next waypoint *nextwp* in the 2-D route is less than *DistStep*, latitude and longitude ($Lat3$ and $Lon3$) of the next point *nextpoint* (which is *DistStep* by distance next to the current

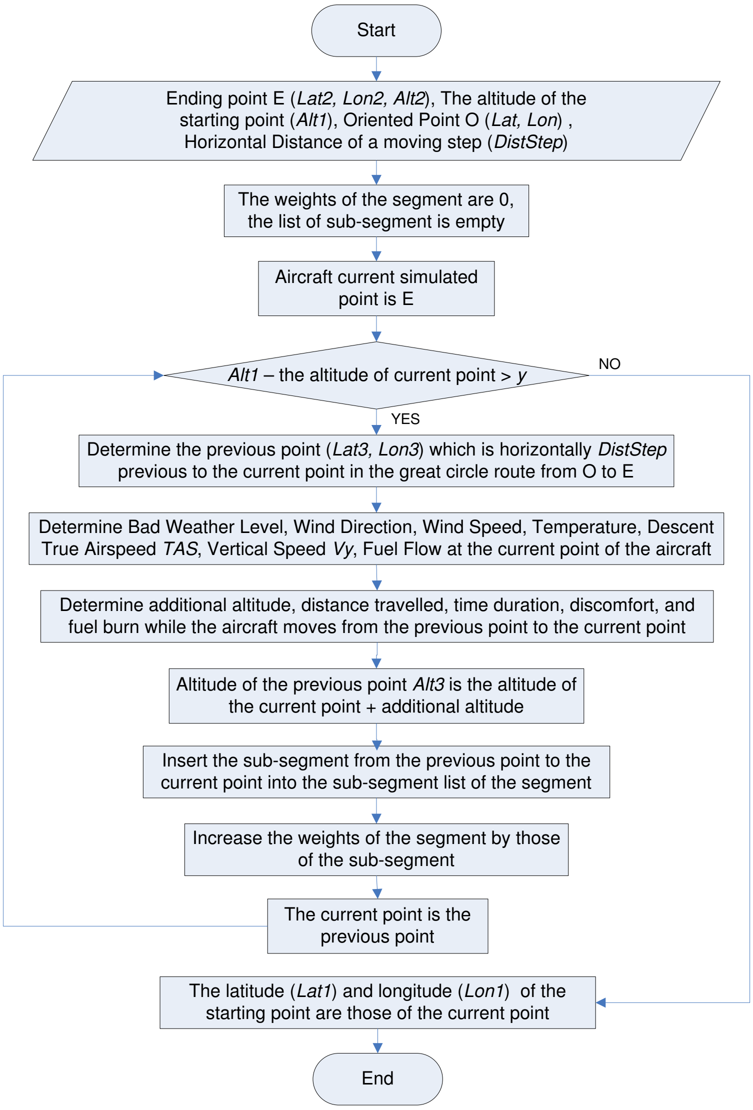


Figure 3.16: Algorithm flow for the backward simulation of aircraft following the great circle route to descend from a starting point given altitude to an ending point

point in the route) are determined. Otherwise Lat3 and Lon3 are the latitude and longitude of the ending point E ($Lat2$, $Lon2$) or the next waypoint *nextwp*.

The weights of the sub-segment from the current point to the next point is calculated by steps as presented in Section 3.7.2. Then the sub-segment from the current point to the next point is inserted into the subsegment list of the segment. The weights of the segment increase by those of the sub-segment.

Figure 3.17 presents the algorithm flow for the simulation of aircraft following a 2-D route to cruise from the starting point S ($Lat1$, $Lon1$, Alt) to the ending point E ($Lat2$, $Lon2$, Alt), where Alt is the cruising altitude.

3.8 Conflict Detection and Resolution

Firstly, we find 3-D (latitude, longitude, altitude) user preferred routes and their 4-D simulated trajectories for flights. Three methods are investigated for doing so: Learning Classifier Systems (LCS), Genetic Algorithm (GA), and Shortest Path Algorithm (Dijkstra) as presented in chapters 5, and 6.

After having found all the 4-D simulated trajectories for flights, conflicts between UPR flights with themselves and non-UPR flights are resolved by changing their given departure times. When a flight is free of conflict, its departure time is called conflict-free departure time. The conflict-free departure times are determined for flights one by one. If a flight does not conflict with flights that have been assigned a conflict-free departure time, this flight will keep its given departure time, otherwise it will increase or decrease its given departure time by a number of minutes. This number of minutes increases by one minute until all conflicts are resolved. If the departure time after decreasing a number of minutes is equal or less than a given time when all flights must take off after that, the departure time will increase only. When a flight is assigned conflict-free departure time, its route's time dimension is updated. We choose this method to resolve conflicts, because the flights routes are better optimised than those when conflicts are resolved during

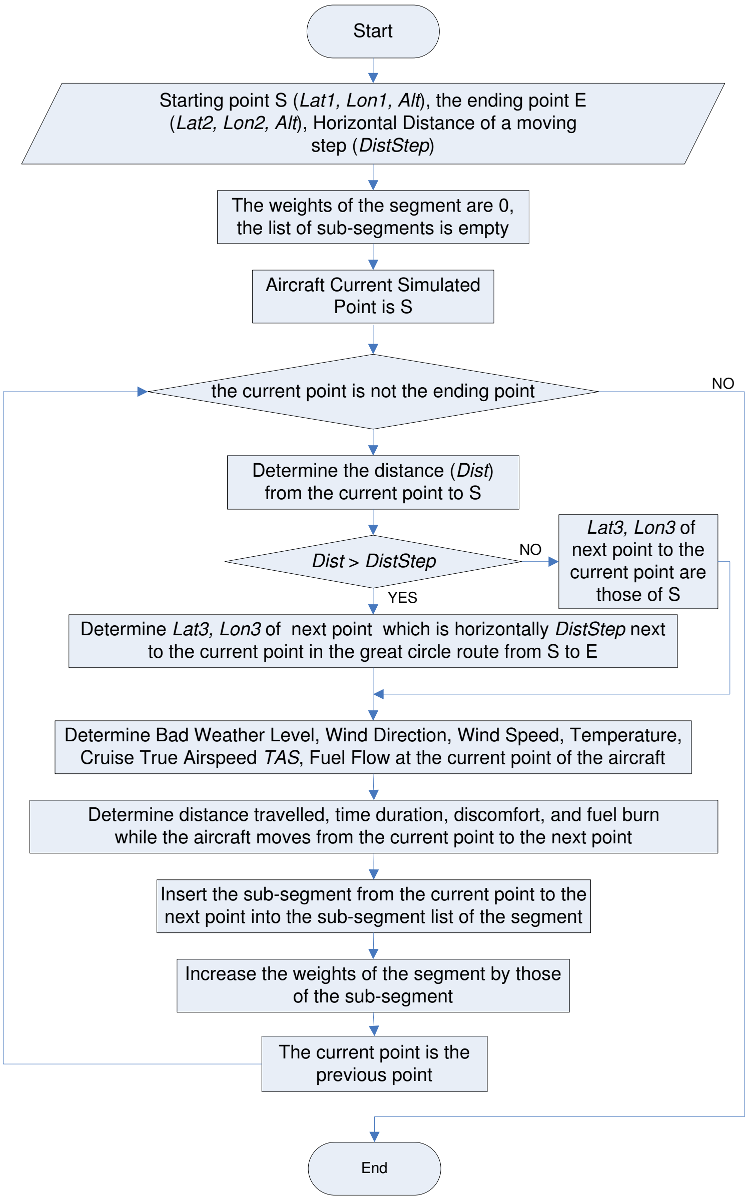


Figure 3.17: Algorithm flow for the simulation of aircraft following a 2-D route to cruise from a starting point to an ending point.

optimisation processes, since the flight routes are deviated from the optimal routes to avoid conflicts.

In order to detect conflict between a flight and the other flights that have been assigned a conflict-free departure time, the following method is used.

The 4-D space is divided into 4-D boxes based on latitude, longitude, altitude and time. The size of a box is 0.5° x 0.5° x 1000 feet x 5 min. The origin of the 4-D space is (-80°, 0°, 0 feet, 0 min). A box will store all the indices of flight segments intersecting with the box, so that a segment only needs to detect conflict with other segments in boxes that both belong to. There has been research dividing the airspace to a grid for detecting conflicts for example in [97]. However the space is divided into 3-D grid (latitude, longitude, time) and if the two segments are in the same box, they are considered to be conflicting with each other. This is also extended to stochastic conflicts: if the probability that traffic in any 3D grid cell is greater than a threshold value, then a resource conflict is flagged.

A list of boxes is maintained. This list is to store all boxes that have at least one intersecting segment. This list is first initialized as empty. When a flight is assigned a conflict-free departure time and its segment indices are added into corresponding boxes, these boxes are also added into the list if they are not already in the list.

Conflict detection between a flight with other flights that have been assigned conflict-free departure time is implemented every time when the departure time of the flight changes. For each segment of the flight, conflict between the segment with all the segments in boxes they both belong to is checked. If the segment conflicts with another segment, conflict detection finishes and a conflict is found. Otherwise, consideration moves on to the next segment. If there is no conflict detected in any segment of the flight, the new departure time will be the conflict-free departure time of the flight. Otherwise, a new departure time is generated by increasing or decreasing the given departure time by a number of minutes.

In order to detect conflict between a segment with another in a box, the following steps are implemented. We assume that the distance of a segment is short enough, so that the velocity vector (defined by bearing, horizontal speed, and vertical speed) is constant in the segment.

The time periods of the two segments are checked to see whether they overlap. Time periods overlap if the arrival time of the starting point of one segment is within the time period of the other. The overlap time (if exist) will start from the latest arrival time (t_1) at the two starting points of the two segments to the earliest arrival time (t_2) at the two ending points of the two segments.

If the two time periods do not overlap, the two flights do not conflict in the two segments. Otherwise the tactical conflict detection algorithm KB3D [47] is used to detect conflict between the two flights. KB3D detects loss of separation and conflict between two flights based on their current state including their position (latitude, longitude, altitude) and velocity vector at a certain time to a lookahead time. Two flights lose separation or violate when their protected zones overlap. The protected zone of a flight is a cylinder of diameter D and height H centered around the flight. In this paper D is chosen as 5 nautical mile and H as 1000 feet. These are standard values of a protected zone [47, 48]. Two flights conflict if they are predicted to lose separation in the future up to the lookahead time. In order to apply KB3D, we need to determine the current state of the two flights at t_1 and then use KB3D to detect loss of separation or conflict between the two flights from their current state at t_1 to a lookahead time $(t_2 - t_1)$. We only need to determine the position (latitude, longitude, altitude) at t_1 of the flight whose arrival time at the starting point is earlier than t_1 as the position at t_1 of the other flight is the starting point of its segment and the velocity vector of the two flights are also provided. Let the time period from the arrival time of the starting point to t_1 be t. The position is horizontally the destination point of the flight traveling along the great circle route from the starting point, given the bearing of the flight and the 2-D distance of the horizontal speed multiplied by t. The altitude of this position is the altitude of the starting point plus the additional altitude of the vertical speed multiplied by t.

3.9 Experimental Framework

In this section, the experimental framework is presented as in Figure 3.18. The framework has a number of blocks. "Artificial Flight Plan" block is to generate UPR and non-UPR flight plans. These UPR flight plans combining with a utility allocation in "User Preference" block results in a flight scenario (UPR flight plans with user preferences) in "UPR Flight Plans with User Pref." block. "Bad Weather Scenarios" block is to generate scenarios of bad weather cells in the airspace with different distributions. "Real Wind Data" block is to process wind and temperature data as presented in Section 3.4. These blocks provide input data for "UPR algorithms" block in general and for "Simulation Environment" in particular. "Simulation Environment" block is to simulate and create networks of control points for "UPR using network" block and to simulate and evaluate 3-D routes for "UPR using GA" and "UPR using LCS" blocks. The fundamental simulations and mathematic for 'Simulation Environment" block are presented in Sections 3.5, 3.6, and 3.7. The application of these to specific problems and UPR algorithms will be presented in Chapters 5, and 6. In this section, we describe "Artificial Flight Plan" block in subsection 3.9.1, "User Preference" and "UPR Flight Plans with User Pref." blocks in sub-section 3.9.2, and "Bad Weather Scenarios" block in sub-section 3.9.3.

3.9.1 Generation of test flights

In order to generate a flight scenario, a single flight plan needs to be generated first. In this thesis, we generate a flight plan as follows.

A table of Origin-Destination-Route-HourOfDepartureTime frequencies is generated from a database of real Australian air traffic data.

The origin, destination, route and hour of departure time of a flight plan is selected by stochastic universal sampling technique based on this frequency table.

Then departure time of the flight plan is completed by sampling randomly a minute from 0 to 55 with the interval of 5 minutes.

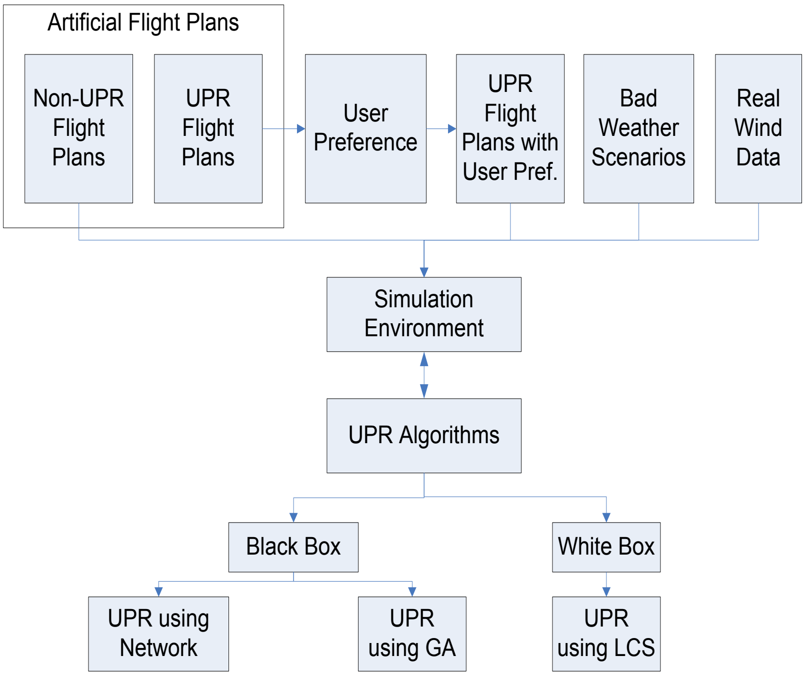


Figure 3.18: Experimental Framework

The aircraft type is selected randomly based on the set of appropriate aircraft types whose maximum range is greater than the great circle distance between the selected origin and destination.

The maximum cruising altitude of the flight is determined by reducing the cruising altitude from the BADA maximum cruising altitude of the selected aircraft by 1000 feet until the cruising altitude is implementable. If the maximum cruising altitude is greater than 10,000 feet then the cruising altitude is selected randomly from 10,000 feet to the maximum cruising altitude. If the maximum cruising altitude is less than 10,000 feet, it will be the cruising altitude of the flight.

Flight plans for 3 consecutive days are generated. Each day has 3000 flight plans. A flight plan is generated as above. Either the route or the cruising altitude is selected, depending on the type of problem (see Section 3.2) to be solved.

Flight plans whose great circle distance between origin and destination is equal or more than 700 km and whose hour of departure time is from 1 PM to 2 PM in the second day are chosen as UPR flights to be optimised. Flight plans are with

Utility Allocation	UP0 - 0.000001 (Discomfort)	UP1 - 0.5 (Neutral)	UP2 - 0.999999 (Time)	Sum
UA0	1	0	0	0.000001
UA1	0.5	0.3	0.2	0.3500003
UA2	0.33	0.33	0.33	0.495
UA3	0	1	0	0.5
UA4	0.2	0.5	0.3	0.5499999
UA5	0.2	0.3	0.5	0.6499997
UA6	0	0	1	0.999999

Table 3.3: Flight utility allocation.

distance longer than 700 km are chosen so that UPR flights can significantly take advantage of wind in enroute phase. For example a flight may choose a longer route instead of great circle route, as the wind in the route can generally provide the ground speed faster than that in the great circle route, resulting in shorter travel time. All other flights are non-UPR flights. However this set of non-UPR flights can be filtered by time dimension so that non-UPR flights need to conflict with UPR flights in time first. With this technique of choosing UPR flight plans, we generated 142 UPR flights and 1,674 non-UPR flights. These 142 UPR flights and 1,674 non-UPR flights are test data for all experiments.

3.9.2 Flight Scenarios

In this thesis, we choose 3 typical user preferences (utilities) to do experiment with: UP0, UP1, and UP2. UP0 is 0.000001, which present the high preference to minimize discomfort. UP2 is 0.999999, which present high preference to minimize time. UP2 is 0.5, which gives neutral preference in minimizing discomfort and time. By choosing the utility of 0.000001, instead of 0 (which gives higher preference to minimize discomfort), flights can choose the minimal time route in the case there are several routes with the same minimal discomfort. If utility is 0, this will not happen, because time factor will be annulled in the objective function by multiplying with utility (0). This advantage is also applied to the utility of 0.999999, instead of 1.

The set of UPR flights is allocated with 7 different utility distributions in Ta-

Utility Allocation	User Pref.	BW0 - 0%	BW1 - 10%	BW2 - 50%	BW3 - 100%
UA0	Time	0	16	70	142
	Neutral	0	0	0	0
	Discomfort	0	0	0	0
UA1	Time	0	11	41	72
	Neutral	0	1	13	43
	Discomfort	0	4	16	27
UA2	Time	0	6	26	47
	Neutral	0	6	24	47
	Discomfort	0	4	20	48
UA3	Time	0	0	0	0
	Neutral	0	16	70	142
	Discomfort	0	0	0	0
UA4	Time	0	3	14	29
	Neutral	0	9	37	72
	Discomfort	0	4	19	41
UA5	Time	0	3	14	29
	Neutral	0	8	27	43
	Discomfort	0	5	29	70
UA6	Time	0	0	0	0
	Neutral	0	0	0	0
	Discomfort	0	16	70	142

Table 3.4: Affected flights in different flight and weather scenarios

ble 3.3 to create 7 different sets (7 scenarios) of UPR flights. In Table 3.3 "UP0 - 0.000001", "UP1 - 0.5" and "UP2 - 0.999999" are the factions of flights with utility of 0.000001, 0.5 and 0.999999, respectively. "Sum" is $0.000001 * UP0 + 0.5 * UP1 + 0.999999 * UP2$. The utility allocation is indexed in the order of "Sum" values.

3.9.3 Weather Scenarios

A scenario of bad weather cells is generated artificially based on a predefined percentage of UPR flights whose great circle routes are affected by bad weather areas. The weather scenario is generated so that it can affect a percentage of UPR flights closest to the predefined percentage. 4 weather scenarios are created, which have the predefined percentage of UPR flights of 0%,10%, 50%, and 100%.

Figure 3.19 is an example of a bad weather scenario which has 9 bad weather cells and affects about 30% of UPR flights. The figure contains two sub-figures. The upper one shows 3-D bad weather cells and the 3-D routes. The lower one show the 2-D (latitude, longitude) routes only to make the 3-D bad weather cells more visible.

Table 3.4 shows the affected flights categorized by user preferences in different

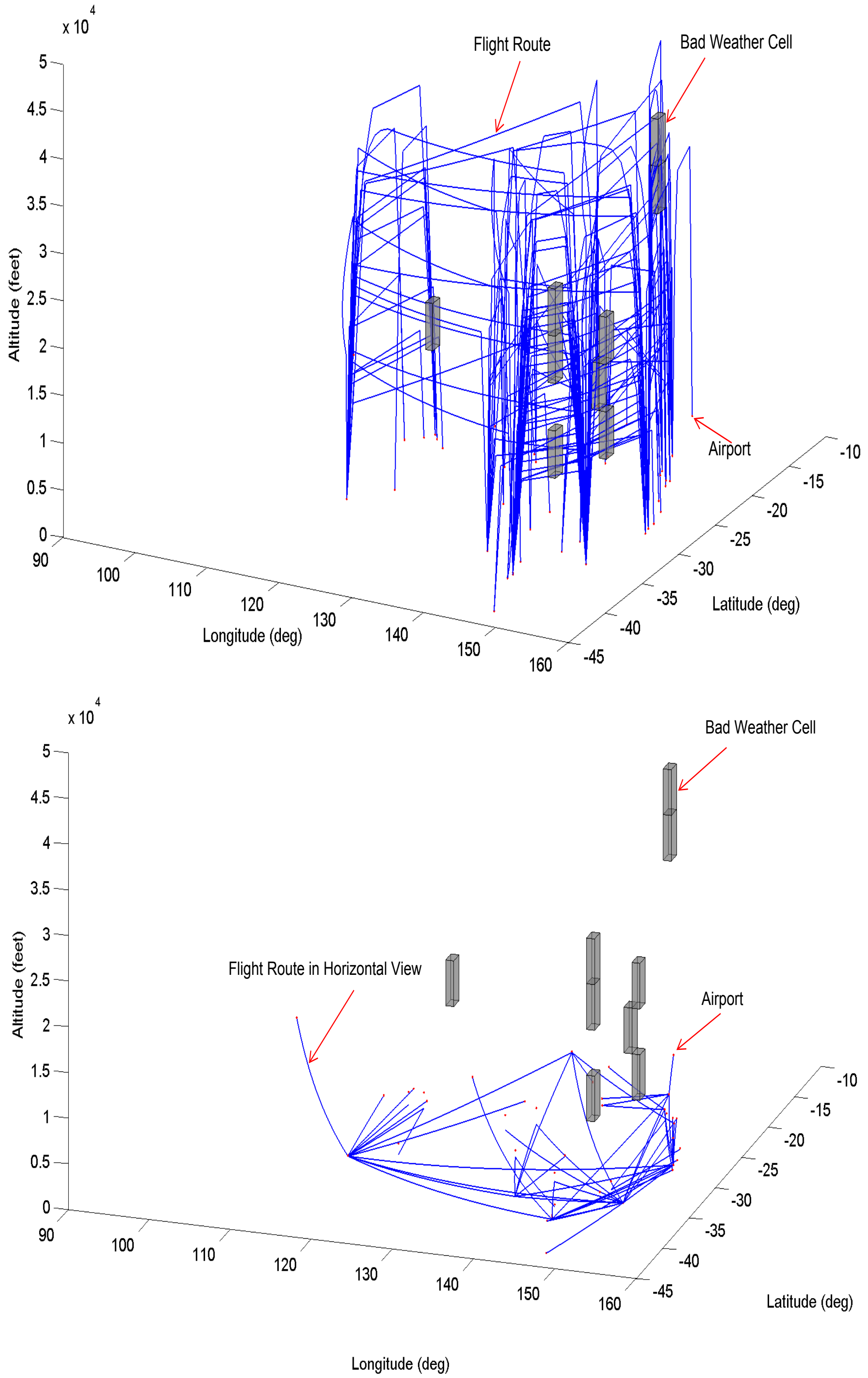


Figure 3.19: An example of bad weather scenario.

weather and flight scenarios. In the table, “Utility Allocation” is the utility allocation for flights from UA0 to UA6; “User Pref.” is the user preference of a flight. “Time”, “Neutral”, and “Discomfort” are three user preferences corresponding to three utility values of 0.999999, 0.5, and 0.000001; “BW0 - 0%”, “BW1 - 10%”, “BW2 - 50%”, and “BW3 - 100%” are bad weather scenarios which affect 0%, 10%, 50%, and 100% of flights effectively.

3.10 Summary

The chapter has set the scene for the experiments described in the rest of the thesis. It described how the UPR problems are formulated, how input data including weather data and flight data are handled, and artificially generated for UPR planning and for UPR algorithms investigation, and how conflicts between UPR flights and non-UPR flights are detected and resolved after having found 3-D user preferred routes and 4-D simulated trajectories of UPR flights. The mathematics and algorithms for fundamental simulation and evaluation of UPR segments and the whole UPR routes are presented. How these settings are applied for a particular problem and a UPR algorithm, and how effectively they work will be presented in the next chapters.

Chapter 4

Aviation Emission Inventory Development and Analysis

This chapter is partially based on following publications:

1. V. V. Pham, J. Tang, S. Alam, C. Lokan, and H. A. Abbass. Aviation Emission Inventory Development and Analysis, Environmental Modelling and Software, vol 25 (12), pp. 1738-1753, ISSN: 1364-8152, Dec, 2010

4.1 Overview

An up to date and accurate aviation emission inventory is a prerequisite for any detailed analysis of aviation emission impact on greenhouse gases and local air quality around airports. In this chapter we present an aviation emission inventory using real time air traffic trajectory data. The reported inventory is in the form of a 4D database which provides resolution of 1° x 1° x 1,000 ft for temporal and spatial emission analysis. The inventory is for an on going period of six months starting from October 2008 for Australian Airspace.

In this study we show 6 months of data, with 492,936 flights (in-bound, out-bound and over flying). These flights used about 2515.83 kt of fuel and emitted 114.59 kt of HC, 200.95 kt of CO, 45.92 kt of NO_x, 7929.89 kt of CO_2, and 2.11 kt of SO_x.

It is expected that with the availability of this real time aviation emission database, environmental analysts and aviation experts will have an indispensable source of information for making timely decisions regarding expansion of runways, building new airports, applying route charges based on environmentally congested airways, and restructuring air traffic flow to achieve sustainable air traffic growth.

4.2 Introduction

Aviation has grown dramatically [117]. Within Australia, air traffic has grown by more than 50% over the last decade. Australia now has approximately 1 million domestic and international flights every year, with up to 4,000 flights on the busiest days. Worldwide air traffic is expected to continue to grow at rates of 3-5% per year [139]. In Australia, it is forecasted to grow by four per cent per annum on average, doubling by 2025. This forecast is based on growth in traveling passengers and the increase in demand for air freight [168].

These levels of growth will mean a considerable increase in aviation traffic and emissions over the next 10 years. Whilst aviation brings considerable economic benefits, this growth is also associated with increased environmental pressures, and consequently a growing need to improve the environmental performance of the industry. Aviation has considerable environmental impacts both at a local airport level and at a regional and global level. Local atmospheric issues are related to airport contributions to local air quality and the potential for health impacts on residential populations in surrounding areas. At a global level, aviation emission has the potential of atmospheric impacts to affect climate. Aviation emissions are increasingly recognized as a significant contributor to global climate impacts through global warming [139]. Models of global warming associated with aviation emissions suggest that aviation contributes approximately 3.5% of the total anthropogenic forcing, and this may increase to between 7-12% by 2050 [139].

Understanding the impact of aviation emissions is important due to the characteristics of aviation emissions occurring in high altitude, and the growth of air traffic.

However, the current state of knowledge of aviation impacts on the atmosphere is severely limited due to unavailability of timely and accurate aviation emission data. New approaches and models are required in order to reduce the considerable uncertainties in emission calculations. Aviation emission inventories are a critical resource because of this requirement.

Various aviation emission inventories have been developed worldwide by the U.S. National Aeronautics and Space Administration (NASA) [178], the U.S. Federal Aviation Administration (FAA) [61] and the European Organisation for the Safety of Air Navigation (Eurocontrol) [59]. Most of these inventories use historic air traffic data, not real time air traffic data. They use several assumptions to simplify the complex flight trajectories, and deal with few representative aircraft types. Moreover the emission inventories are not comprehensive i.e. they lack details regarding emission in different flight phases (such as taxi-out and taxi-in). Therefore the results obtained are not very accurate, and detailed, and by the time they are compiled they become outdated.

In this chapter we present an aviation emission inventory methodology that is generic in nature, but applied to Australian Airspace only because of data availability. The methodology uses real time air traffic trajectory data and real-time computation of aviation emissions. We have developed software called TOP-LAT (Trajectory Optimisation and Prediction of Live Air Traffic), which constructs the trajectory of a flight based on real time flight plans, routes and radar signatures of the flight. The fuel flow and emissions of the flight are calculated every time step using the speed, altitude, and phase of the flight, along with aircraft-specific aerodynamic models. These results are combined with aircraft position and time to generate a 4D (latitude, longitude, altitude and time) emission database. With these characteristics, the model provides more accurate fuel flow and emissions computation than the other models.

We also present a generic temporal and spatial analysis methodology of the impact of air traffic on the environment, with a case study on the Australian airspace. Furthermore, spatial distribution and analysis of aviation emissions including HC, CO, NO_x, CO_2 and SO_x are presented.

4.3 Environmental Impact of Aircraft Emissions

Typically, aviation emission consists of 71% Carbon dioxide (CO_2) and 28% water vapour (H_2O). In the remaining 1%, NO_x is the most important emission [139, p. 28]. Four main factors [94] influencing aviation emissions include

- the intensity and volume of aircraft movements
- the type and spatial concentration and distribution of the particular pollutants
- fuel consumption and energy efficiency
- and the rate of renewing of the aircraft fleet by introducing “cleaner” aircraft.

The effects of aviation emission are classified in the literature as long-range air pollution, effect on the ozone layer, and greenhouse effects [94, 28]. They are presented in more detail in the following subsections.

4.3.1 Long-range air pollution

Long-range air pollution refers to the effects of air pollution far away from the polluting source. One such effect is “acid rain” caused partially by SO_x and mainly by NO_x produced by air traffic [28]. Acid rain has harmful effects on plants, aquatic animals, and infrastructure. Acid rain is mostly caused by human emissions of sulfur and nitrogen compounds which react in the atmosphere to produce acids.

4.3.2 Effect on the ozone layer

When aircraft-related pollution appeared, complex chemical reactions were initiated. The NO_x from aviation emission in the troposphere (where subsonic flights take place) has enhanced the ozone concentration [94]. Tropospheric (ground level) ozone is a damaging oxidant. It is also considered as one of the greenhouse gases in the troposphere. The concentration of NO_x in the upper troposphere has been large, and during winter months aircraft have been identified as dominant sources of the NO_x at locations in the Northern Hemisphere. At higher altitudes in the stratosphere (where supersonic flights have taken place), an increase in concentration of NO_x has depleted the ozone layer [139] which absorbs much of the negative ultraviolet sunlight waves that would otherwise reach the surface of the earth and affect the biosphere. The crossover surface between production and destruction of the ozone layer lies roughly between the lower stratosphere and high troposphere — the tropopause, where long-range aircraft cruise [197]. Aviation-induced nitrogen oxides (NO_x) make up 2% of all human generated NO_x emissions [139]. NO_x has 10 times higher residual time in the tropopause than at ground level.

4.3.3 Greenhouse effects

CO_2, H_2O emitted by aircraft absorb thermal infrared radiation, emitted by the Earth's surface and by the atmosphere. This causes global warming of the earth [92]. The aviation sector contributes 1.6–2.2% to all human generated CO_2 emissions [163]. H_2O emissions from aviation are small compared to the water evaporating at the Earth's surface. H_2O resulting from aviation emission is mainly emitted into the troposphere, where it is removed by precipitation. Besides that, contrails also tend to warm the Earth's surface, similar to thin high clouds [139]. The condensation trails are triggered from the water vapor emitted by aircraft and their optical properties depend on the particles emitted or formed in the aircraft plume and on the ambient atmospheric conditions. Aircraft line-shaped contrails are estimated to cover about 0.1% of the Earth's surface on an annually averaged basis with larger regional values. The contrail cover is projected to grow to 0.5% by 2050.

4.4 Aviation Emission Inventories

There are two main approaches in the literature for developing aviation emission inventory. One is on a global basis for determining the impact of aviation emission on climate change, and the other is on a local basis, which focuses on the level of air quality surrounding an airport. Emissions of carbon dioxide (CO_2) and water vapour (H_2O) are of particular interest in global warming, whereas investigations of hydrocarbons (HC) and carbon monoxide (CO) are done for their effects on air quality.

Emission inventories for global aviation are available from different sources, most prominently the U.S. National Aeronautics and Space Administration (NASA), the European Community (EC) Working Group, and FAA.

The NASA study, commonly known as "Scheduled Civil Aircraft Emission Inventories for 1999" [18, 178], was done by the Boeing Group. Three-dimensional data of aircraft fuel burn and emissions of NO_x, CO and HC were determined for the year 1999.

The AERO2K emission inventories [117] were developed by the European Community in a common project by QinetiQ, Eurocontrol, Manchester Metropolitan University, the National Aerospace Laboratory (NLR) of the Netherlands and the German Aerospace Centre (DLR). Emissions of CO_2, H_2O, NO_x, CO and HC were calculated along with the fuel-use and distance travelled.

The System for Assessing Aviation's Global Emissions (SAGE) [62] has been developed by FAA. Currently SAGE is a research tool and not available in the public domain. However, preliminary results of the model are available.

All the models are used to estimate the amount and the distribution of different emissions from aviation. All of them are designed to calculate aircraft emissions over the world's airspace by latitude, longitude, and altitude. Historical inventories of aviation emissions and future trend forecasts have been produced by these models.

		NASA	AERO2K	SAGE	This study
General information	Years	1976, 1984, 1992, forecast for 2015	2002, forecast for 2025	2000 - 2004, forecasts in development	Real time emission database 2008 onwards
	Coverage	Scheduled aviation, Charter aviation, the former Soviet Union/China unreported flights, Military aviation	Scheduled and unscheduled aviation, Military aviation	Scheduled and unscheduled aviation	Scheduled and unscheduled aviation
	Working mode	Off-line	Off-line	Off-line	On-line
	Processed data	Historical data	Historical data	Historical data	Real time data
Movements data	Sources	OAG flight schedules	ATC data from ETMS & Air Traffic Flow Management Modelling Capabilities (AMOC), BACK flight schedules	ATC data from ETMS, OAG flight schedules	EUROCAT (Thales Radar Data Processing Software) radar data, Flight plans
	Data collection period	Each month of 1992, 4 months of 1976 and 1989	6 representative weeks of 2002 (ATC), Each month of 2002 (schedules)	Each month of 2000 - 2004	Real time data
	Contents of database	Scheduled information (no waypoints/trajectories)	4D trajectories from ATC data, Artificial routing for scheduled flights	4D trajectories from ATC data, Dispersed great-circles for scheduled flights	Flight plan, Ground Movement Report, 4D Automatic Dependent Surveillance – Broadcast (ADS-B) fused Radar track data
Performance	Representative Aircraft/Engine (AC/Eng) combinations	27 / 36 / 76 for 1976 / 1984 / 1992 (jets and generic turboprops)	40 (jets and turboprops)	91 (jets and turboprops)	91 (jets and turboprops) mapping to 200 aircraft
	Performance data, Performance model	Boeing proprietary data, Boeing Mission Analysis Program (BMAP)	PIANO aircraft models, PIANO performance software	BADA & Integrated Noise Model (INM) aircraft models, BADA & INM performance methodologies	BADA & aerodynamic performance model
	Selected mission assumptions	Great-circle routes, 70% passenger load factor, Continuous climb cruise	Real routing, 60.9% load factor (by mass), Cruise with step climbs	Real routing for ETMS flights, Take-off weight estimated from INM, Cruise with step climbs, Delay modelling	Real routing from flight trajectory data, actual flight mission plan, airport-runway-aircraft specific Taxi-in and Taxi-out Time, nominal aircraft weight with fuel burn reduction.
Emissions	Emission data, Emission model	ICAO emission indices + industry data, Boeing-2 fuel flow method (NO_x, CO, HC)	ICAO emission indices + industry data, DLR fuel flow method (NO_x), DLR Omega method (CO, HC), DLR Soot method (particles)	ICAO emission indices + Emissions and Dispersion Modeling System (EDMS) data, Boeing-2 fuel flow method (NO_x, CO, HC)	ICAO emission indices + industry data, Boeing-2 fuel flow method (HC, CO, NO_x, CO_2, SO_x)
Assumptions	Atmosphere & wind	ISA atmosphere and zero wind	ISA Atmosphere and zero wind	ISA Atmosphere and zero wind	Actual atmosphere and actual wind
Results	Allocation software	Boeing-developed Global Atmospheric Emissions Code (GAEC)	AERO2k data integration tool	SAGE fuel burn and emission module	TOP-LAT
	Species covered	Fuel burn, NO_x, CO, HC	Fuel burn, CO_2, H_2O, NO_x, CO, HC, soot + distance per grid cell	Fuel burn, CO_2, H_2O, NO_x, CO, HC, SO_x + distance per grid cell	Fuel burn, HC, CO, NO_x, CO_2, SO_x
	Resolution	3D data in a 1° x 1° x 1 km world grid	3D data in a 1° x 1° x 500 ft world grid, 4D data in a 1° x 1° x 500 ft x 6h grid	3D data in a 1° x 1° x 1 km world grid, 4D raw data at any resolution required	3D data in a 1° x 1° x 1 kft grid, 4D raw data at any resolution required

Table 4.1: State of the Methodologies in Aviation Emission

There is a common approach in all of the aforementioned 3D emissions models. All of them operate in off-line mode. They all involve combining a database of global air traffic (fleet mix, city-pairs served, and flight frequencies) with a set of assumptions about flight operations (flight profiles and routing) and a method to calculate altitude-dependent emissions of aircraft/engine combinations. Idealized flight routings and profiles, with no winds or system delays are assumed. Thus there can be a vast discrepancy between actual fuel burn and emissions and estimated fuel burn and emissions using idealized scenarios.

The trajectory of a flight in these models is also not very accurate. It is modelled via a number of waypoints. For example, SAGE models the trajectory of a flight by 30–40 chords, AERO2K uses 20–30 waypoints to represent the trajectory. This affects the accuracy of fuel and emission computation.

In the emission inventory developed by NASA, aircraft emissions of CO_2, H_2O and SO_2 were determined directly from fuel consumption. Since great-circle routes were assumed between city-pairs, the results are more suited for trend analyses. A major drawback, however, is the incomplete coverage of global aviation, since only scheduled air traffic was accounted for. Furthermore, the output data does not contain any four-dimensional results (emission data with time) limiting any seasonal and temporal trend analysis.

In AERO2K, the generation of a movement database from Air Traffic Control (ATC) data is highly time-consuming and cumbersome, due to legacy systems and different data formats, and can only be done with historic air traffic data.

Table 4.1 provides a comparative analysis of the above mentioned state of the art aviation emission inventories, together with our methodology which is used to develop the emission inventory. This table is developed based on [158] and compares our methodology with others in the literature.

The main advantage in our study is that the system is on-line and processes air traffic data in real time, capturing not only effects of wind & flight delays (holdings) but also complex vectoring maneuvers during approach by Air Traffic Controllers

(ATCs). The 4D emission database is updated in real time, giving a live picture of emission both for local air quality as well as greenhouse impact. The system uses actual 4D flight trajectories, thus it provides a more accurate estimate of fuel burn and emissions.

4.5 Australian Emission Inventory Development

The emission inventory is in the form of a relational database which records emissions data in four dimensions. Latitude, longitude and altitude divide Australian Airspace into a 3D grid; the fourth dimension is time. The dimensions of the grid are the National Airspace's start latitude, start longitude, end latitude, and end longitude, with altitude ranging from ground level to 61,000 ft. The size of one cell in the grid is 1 ° (longitude) x 1 ° (latitude) x 1,000 ft (altitude). This can be changed to increase or decrease the resolution of the grid. Fuel and emissions are calculated for all of the grid cells as and when the flights transit through them.

Calculation of fuel and emissions is based on accurate flight trajectories, constructed from real time data on latitude, longitude and altitude at a number of intervals throughout each flight. This allows appropriate fuel and emission data for the actual flight route and altitude to be estimated. Computing fuel and emissions from real time radar data is a significant improvement over previous models. Previous models usually assume a great circle route between origin and destination airports and also a typical cruise altitude. There can be a large difference between the actual route flown and the great circle route, due to air traffic control (congestions, flight corridor), restriction on airspace (SUAs), weather (turbulence, storm), or avoiding/taking advantage of prevailing winds.

There are two estimates available for the aviation fuel and emissions in Australia. U.S FAA's System for assessing Aviation's Global Emissions (SAGE) estimated the global aviation fuel and emissions from 2000 to 2004 and National Greenhouse Gas Inventory Committee (NGGIC), Department of Climate Change, Australia from 1990 to 2007. However, SAGE is a global aviation emission inventory where the focus is on aggregated numbers for fuel burn and emissions using flight plan data alone for Australian Airspace. NGGIC's estimation is undertaken using a Tier 2 methodology [91] and was based on aviation fuel sold rather than flight trajectories or even flight plans [131]. Therefore, these estimates provide only an aggregate number for fuel burn and aviation emission in Australia. The aviation emission inventory developed in this chapter is not only a step ahead (for it uses actual aircraft trajectory data) but also attempts to quantify data from spatial and temporal perspectives.

4.5.1 Proposed Methodology

The essence of our approach is that radar, flight plan and aircraft specific aerodynamic models are used to predict and construct accurate flight trajectories. A flight trajectory consists of a set of chords where the start and end of the chord represents a valid radar position for that flight. This can be as small as 30 seconds or as long as 2 hours depending upon radar coverage. For every chord the fuel flow and emission is computed by forward integrating the aircraft position over time using the aerodynamic model for that aircraft and its associated engine. This allows for fine-grained computation of fuel flow and emission at every time step between two valid radar signatures. Figure 4.1 illustrates the computation process used for developing the aviation emission inventory. It consists of four main components, described in the following sections.

4.5.2 Real time Data Processing

Real time Data Processing performs the assimilation of flight plan, flight route and radar data; structuring of air traffic and flight plan data; elimination of military, helicopters and recreational flights; assigning flight type (over flying, inbound, outbound, inland); aircraft mapping; aircraft-engine assignment; runway allocation; estimating time of departure; and building the initial flight plan and route profile. This component continuously updates the initial flight plan information regarding

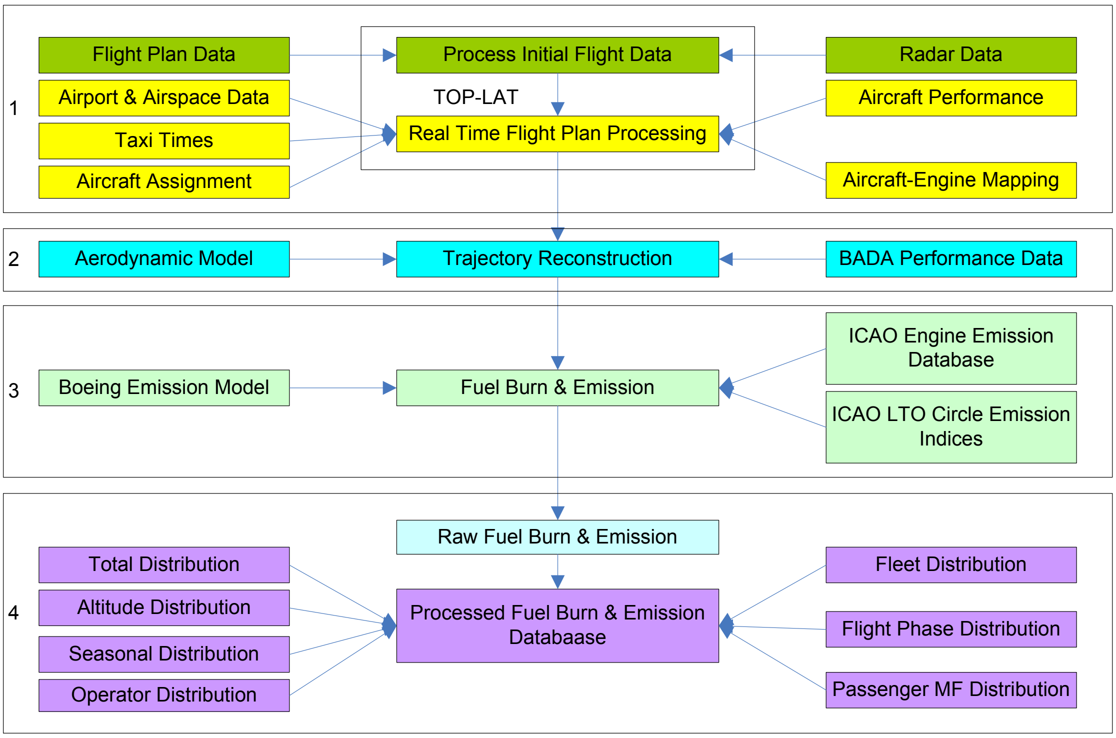


Figure 4.1: Aviation Emission Inventory Development Process

the runway, arrival and departure route as data for the respective flight is received.

Air Traffic Control (ATC) data from the EUROCAT radar system is used for generating flight trajectories. Both flight plan and trajectory information from the EUROCAT radar data need to be aligned in space and time, get processed for data integrity, data validation, and mining data connection. As a consequence, checks and assessments are required before the data can be used for baseline inventory production. The processing of flight plan and radar data includes the substitution of missing data (departure route, arrival route, departure runway, arrival runway, cruising altitude, aircraft performance data) from other available sources, and the identification and removal of inconsistent or redundant information.

The system focuses on civilian air traffic only. Some flights, such as small recreational flights and helicopters, are not processed by the system as aircraft/engine data is not available yet. Military flight data is also not taken into account because they are classified. The amount of emissions produced by these flights is relatively small when compared with commercial flights.

4.5.3 Trajectory prediction and reconstruction

This is mainly performed by a system called Trajectory Optimisation and Prediction of Live Air Traffic (TOP-LAT), developed by the authors [1]. The system reconstructs the trajectory of a flight based on information about the flight including flight plans, flight routes (the system can receive several flight plans and flight routes of the flight; representing different updates for the flight plan) and all radar data of the flight.

The trajectory prediction component of TOP-LAT predicts the aircraft trajectory from one radar point to the next radar point. When it reaches a radar point it computes the time of arrival at the next waypoint and computes the acceleration/deceleration rate as well as the climb/descent profile to meet the next radar point requirements. These computations are based on the aircraft's current position, altitude, speed and aircraft performance parameters. The trajectory reconstruction component of TOP-LAT updates the flight trajectory for flights that have missing/incomplete radar data. It then forms the completed flight plans which are compliant with the flight plan structure. The radar data positions are converted into the route of the flight plan with necessary modifications (considering distances, heading, speed, and time between two consecutive radar signatures).

Trajectories based on EUROCAT radar data often need to be smoothed, i.e. kinks and altitude spikes need to be removed. Trajectory kinks may result from the limited accuracy of radar stations, if closely spaced waypoints originate from different radar centers. Similarly, altitude spikes are identified and smoothed by defining a maximum rate-of-climb (ROC) value between successive waypoints.

To determine the taxi-in time of a flight, firstly the departure route of the flight is estimated, based on the flight's current trajectory which includes waypoints and

[1] Involved people include: H.A. Abbass, S. Alam, A. W. Iorio, C. Lokan, V.V. Pham and J. Tang in alphabetical order. My role includes: Develop tools to process and integrate air traffic data such as flights, airports, waypoints, SIDs, STARs and airspace sectors data; develop a tool to predict flight trajectories, track and visualize flight trajectories; develop a tool to query and analyse air traffic data fuel, and aviation emissions; develop wind and atmospheric data processing and integrating tool; develop User Preferred Trajectory tool

radar signatures, the aircraft type, the operator of the flight and the matrices of departure route probability by waypoints, destination, operator, and aircraft. Then the departure runway is estimated based on departure route, the destination, waypoints in the trajectory, and departure runway frequency matrices by destination, waypoints, and aircraft type. If arrival runway and taxi-in time of the arrival runway is available in the database, taxi-in & taxi-out time of the flight is determined, else a default value is assigned for taxi-in time. The database contains taxi-in times for all major airports in Australia.

The trajectory of a flight in the system is constructed based on the actual flight plan and actual radar data. Therefore it is more accurate than a trajectory that is based upon a single great circle route in other inventory and modeling studies such as those by NASA/Boeing [178].

4.5.4 Fuel flow and emission computation

4.5.4.1 Aircraft Performance and Fuel Flow

Fuel flow is computed between two consecutive radar positions by forward integrating the aircraft position over time, using the aerodynamic parameters of the specific aircraft. The 91 supported aircraft types from BADA database are mapped to another 200 aircraft to deal with aircraft types which are not available in BADA database.

For each position of the aircraft in a discrete time interval of 1 second, the fuel flow is computed based on BADA thrust specific fuel consumption model as in Section 3.6.2. The absolute amount of fuel burn in a flight segment is calculated by multiplying fuel flow with time.

The effect of wind on aircraft during flight is intrinsically incorporated as they impact directly on flight time. The real take-off weight of an aircraft is not provided by airlines, thus the nominal mass, as suggested by ICAO, is used. It is around 60.9% of maximum take-off weight (ICAO average value for scheduled air traffic).

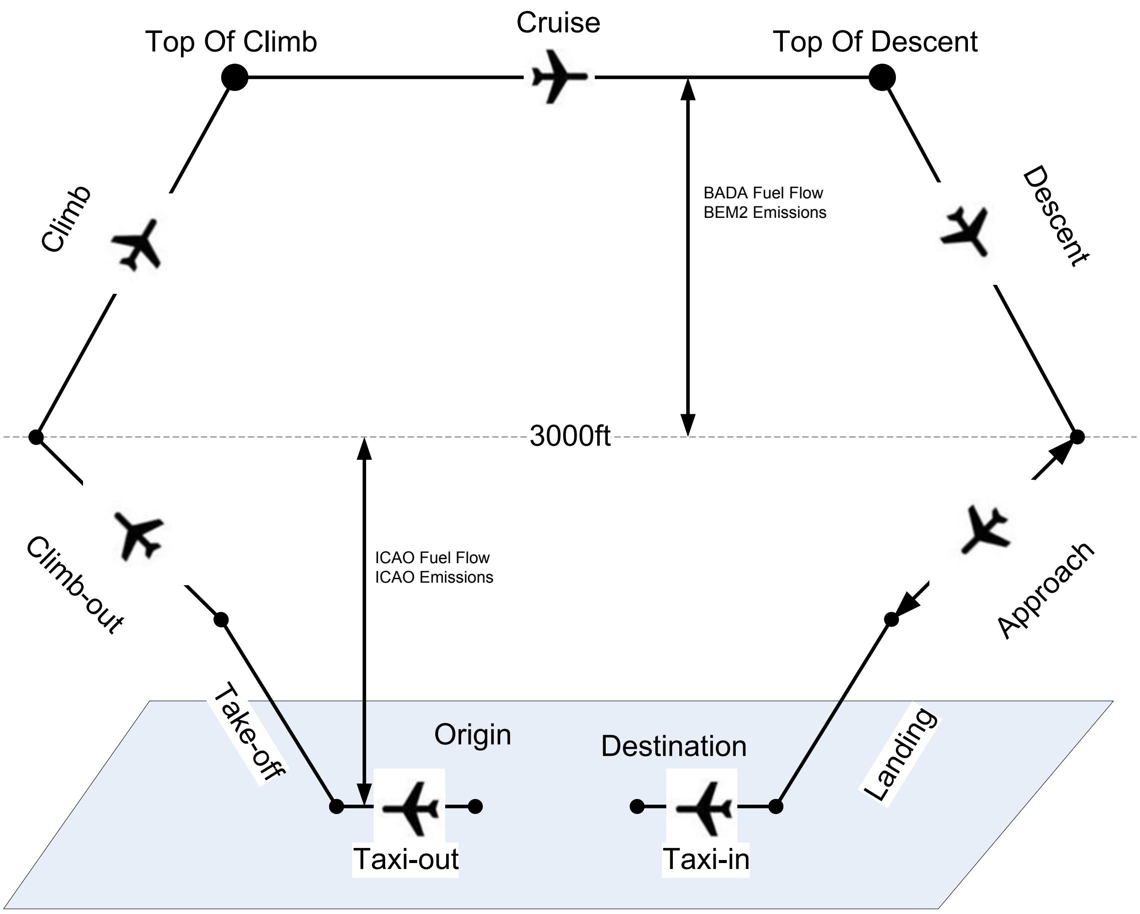


Figure 4.2: Emission computation phases.

The weight of the aircraft is reduced as a result of fuel burn while it is flying. This is done by subtracting aircraft take-off weight with the instantaneous fuel burn.

4.5.4.2 Emission computation

Emission is calculated using Boeing Emission Method 2 (BEM2) methodology and ICAO Engine Exhaust Emissions Data Bank. SO_x, NO_x, CO, HC and CO_2 emissions are computed for each flight.

The emission computation process is divided into two phases, below 3000 ft and above 3000 ft, as shown in Figure 4.2.

Below 3000 ft, the emission calculation is based on the ICAO Engine Exhaust Emissions Data Bank [86]. The fuel burn calculation is based on the Landing and Take-Off Cycle (LTO) defined by the ICAO Engine Certification specifications. ICAO LTO covers four engine operation modes, which are used to model the six phases of aircraft operations. In order to calculate fuel and emissions for an aircraft/airframe type using the ICAO data bank, we firstly find the engine corresponding to the aircraft type by using Airplane/Engine Substitution tables as used in the NASA study [178, Appendix B]. Then we use the ICAO Engine Exhaust Emissions Data Bank for the given engine to calculate fuel and emissions for a flight while it's in idle, approach, climb-out and take-off states. It can be seen that in the ICAO Engine Exhaust Emissions Data Bank, the HC and CO emission indices of jet engines in idle state are higher than in the other states [86]. The emission and fuel data is given for single engine, it is then multiplied by the number of engines depending upon the type of aircraft it is fitted on.

In this study the time-in mode is derived from actual flight trajectory data, and runway timings are specific for each airport and aircraft type. This gives the emission computation a high level of fidelity as compared to using generic ICAO LTO cycle timings.

Above 3000 ft, the emission calculation is also based on the ICAO Engine Exhaust Emissions Data Bank, but emission factors are adopted to the atmospheric conditions at altitude by Boeing Emission Method 2 (BEM2) [18, 62], which calculates emission indices based on fuel flow and ICAO certification data. ICAO data at the four certified power settings at sea-level conditions are used to compute the resulting emissions, while correcting for atmospheric conditions. BEM2 allows for the estimation of emissions for pollutants such as NO_x, HC, and CO. The emission for CO_2 is directly proportional to fuel burn with CO corrections. ICAO's Committee on Aviation Environmental Protection (CAEP) has recommended the adoption of Boeing Method 2 as a standard method for calculating emissions [88].

BEM2 method for emission computation is described here as detailed in [18, 50]. BEM2 computes flight emissions using, as a base, the measured fuel flow and the engine ICAO data sheets. It accounts for ambient pressure, temperature and humidity as well as Mach number.

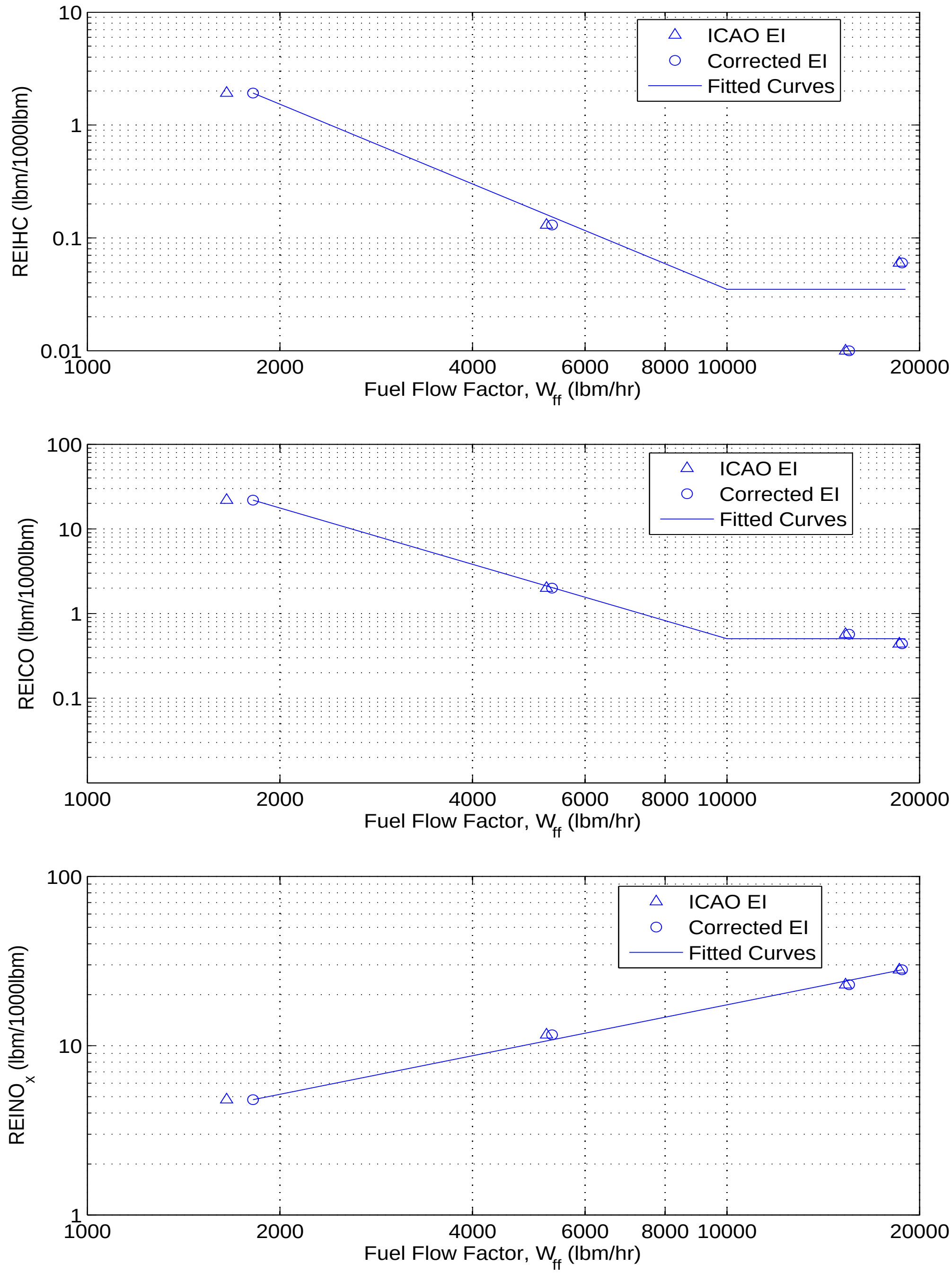


Figure 4.3: Reference EI and Fuel Flow Factor.

The ICAO data sheet contains complete information about the fuel flow, the Hydrocarbons (HC), the Carbon Monoxides (CO), and the Oxides of Nitrogen (NOx) for the four power settings. These settings are 7%, 30%, 85%, and 100%.

BEM2 steps of emission computation are as follows:

1. Curve fitting the ICAO engine emission data

   The fuel flow is converted from $\frac{kg}{s}$ to $\frac{lbm}{hr}$ by multiplying by 7936. The emission index (EI) values will not change but the units are $\frac{lbm}{1000lbm}$

   The fuel flow given does not account for the installation effects of engine air bleed for aircraft use so a correction must be made. The adjusted EI is defined as the reference EI or REI. The correction rates for the four power settings are 1.100, 1.020, 1.013, and 1.010 respectively.

   Once the conversions and corrections are made, the emission indices (EI) are plotted against the corrected W_f on $log_{10} - log_{10}$ scale as in Figure 4.3. The data points are curve-fitted to show trends of EI for different fuel flows.

   The HC and CO are bi-linear least square fitted curves. The NO_x curve is a simple point-to-point linear fit, on the log_{10} paper, between the ICAO emission data points.

2. Compute fuel flow factor

   (a) Calculate the values ∂_{amb} (ambient pressure correction factor) and θ_{amb} (ambient temperature correction factor) where:

   - $\partial_{amb} = \frac{P_{amb}}{14.696}$ (P_{amb} = ambient(inlet) pressure) and
   - $\theta_{amb} = \frac{T_{amb}+273.15}{288.15}$ (T_{amb} = ambient(inlet) temperature)
   - $\theta_{boiling} = \frac{T_{amb}+273.15}{373.15}$

   (b) The fuel flow values are further modified by the ambient values and are denoted by W_{ff}:

   - $W_{ff} = [W_f/\partial_{amb}]\theta_{amb}^{3.8}e^{0.2M^2}$ ($\frac{lbm}{hr}$) where M is the Mach number.

3. Compute Emission Indices (EI)

   Calculate the emission indices of HC, CO and NO_x:

   - $EIHC = REIHC\frac{\theta_{amb}^{3.8}}{\partial_{amb}^{1.02}}\ (\frac{lbm}{1000lbm})$
   - $EICO = REICO\frac{\theta_{amb}^{3.8}}{\partial_{amb}^{1.02}}\ (\frac{lbm}{1000lbm})$
   - $EINO_x = REINO_x e^H \langle\frac{\partial_{amb}^{1.02}}{\theta_{amb}^{3.3}}\rangle^{(1/2)}\ (\frac{lbm}{1000lbm})$

   where $REIHC$, $REICO$, $REINO_x$ = intersection of corresponding curves and W_{ff}, and the humidity correction factor H is calculated as follows

   - $H = -19.0(\omega - 0.0063)$, ω = specific humidity assumed to be 60% for the entire flight duration.
   - $\omega = \frac{0.62198\times\Phi\times P_v}{P_{amb}-\Phi\times P_v}$, where Φ is relative humidity and P_v = saturation vapour pressure in psia (Pounds per square inch absolute (including atmospheric pressure)) where $P_v = 0.014504 \times 10^{\beta}$
   - $\beta = 7.90298(1-\frac{1}{\theta_{boiling}})+3.00571+5.03808log(\frac{1}{\theta_{boiling}})+(1.3816\times10^{-7})[1-10^{11.344(1-\theta_{boiling})}] + (8.1328 \times 10^{-3})[10^{3.4919(1-\frac{1}{\theta_{boiling}})} - 1]$

4. Total Emission

   Assume that a flight segment has n sub-segments (moving steps). The fuel flow factor and time travelled at sub-segment i are W_{f_i} and $time_i$ respectively. The corrected fuel flow factor W_{ff_i} is calculated by Step 2. $EI(HC, CO, NO_x)_i$ are then calculated by Step 3. The total amount of emissions for the segment is computed by

   $S(HC, CO, NO_x) = Number\ of\ Engines \times \sum_{i=1}^{n} EI(HC, CO, NO_x)_i \times W_{f_i} \times time_i \times 10^{-3}$ (lbm)

   Then $S(HC, CO, NO_x)$ are multiplied by 0.45359237 to convert to kg.

   CO_2 emission methodology [59, p. 32]: Carbon dioxide and water are the main products of the combustion of hydrocarbon fuel and are therefore a function of fuel flow. One kg of Jet fuel produces 3156 g of CO_2 and 1237 g of H_2O [59, p. 31]. For accurate computation of CO_2, partially or unburnt species of CO and

HC must be subtracted. Since the amount of hydrocarbons emitted is less than 1% of CO_2 and H_2O emissions and their hydrogen-carbon ratio is unknown, they are neglected in our computations. The formula for CO_2 computation is given by

- $EICO_2 = EICO_{2,ideal} - (44/28) \times EICO$, where $EICO_{2,ideal}$ is 3156 (grams CO2/kg fuel burned) and 44/28 ratio represents their different molar masses.

SO_x emission methodology [59, p. 64]: The maximum sulfur content in kerosene according to international regulations is 0.3 mass percent with actual values often below this limit. For a sulfur content of between 0.001 and 3 g per kg fuel one can expect SO_x production during combustion of 0.6-1 g/kg. The formula for SO_x computation is given by:

- $SO_x = 0.8 * W_f$ (kg)

This value is based on [75] and was used in the NASA and SAGE inventories.

4.5.5 Emission Database

The database acts as a dynamic repository of flight state information and emission data. The database is divided into two main parts: flight database, and gridded emission database.

The flight database stores all the flight-related information (e.g trajectory, flight plan).

The emission database stores aggregated fuel burn and emissions per grid cell. Lambert Canonical Projection system is utilized based on longitude and latitude; these are supplemented by an altitude coordinate which is oriented orthogonally to the earth's surface. Based on this coordinate system, the three-dimensional airspace is divided into cells, which span the whole Australian National Airspace from ground level up to a maximum altitude of 61,000 ft. Fuel burn and emissions of each flight are calculated and allocated to a grid cell. For the purpose of emission allocation, the intersections of the flight path with the grid cell boundaries are determined by their three dimensional coordinates. The process is repeated for all flights processed from the radar data, and the respective emissions per grid cell are summed.

From the database, we can query the emission data in any area at any time. Emission data can also be presented with any required resolution by the system.

4.5.6 Software and Data Availability

This chapter presents the aviation emission inventory developed by using Trajectory Optimisation and Prediction of Live Air Traffic (TOP-LAT) software. However, the emission database is owned by Airservices and TOP-LAT is currently being commercialized. TOP-LAT is developed in Microsoft Visual C# and it is based on a distributed client-server architecture. The main server processes real time air traffic data, and other sub-systems compute fuel, emission, safety, etc.

The system provides a comprehensive emission inventory in the form of emissions and fuel data as well as flight specific information which is updated by TOP-LAT and stored in a 4D gridded database in real time which represents national airspace in the form of a 4D grid, representing the latitude, longitude, altitude and time which can be used for spatial and temporal analysis of emission.

This emission inventory provides valuable information on the quantity and location of emissions in the Australian National Airspace. The previous international studies only used published flight plans and assumed a great circle route between origin and destination airports. The use of radar data has considerably enhanced the knowledge and accuracy of the actual position (latitude, longitude and altitude) at which these emissions actually occur. This provides Air Navigation Service Providers with a higher fidelity and more accurate estimate of aviation emissions. This emission database provides a major new data source for aviation climate impact assessment.

From the database, we can query the emission data in any area at any time.

	Flights	Distance	Fuel	HC	CO	NO_x	CO_2	SO_x
Total	492,936	295,303	2515.83	114.59	200.95	45.92	7929.89	2.11
Domestic	83.86%	58.61%	41.68%	96.57%	96.34%	32.37%	41.68%	41.68%
International	16.14%	41.39%	58.32%	3.43%	3.66%	67.63%	58.32%	58.32%

Table 4.2: Flight movements, distance traveled, fuel used, and emissions of domestic and international Flights in Australian National Airspace during the 6-month study period. Distance is measured in knm, and fuel and emissions in kt.

Emission data can also be presented with any required resolution by the system.

4.6 Aviation Emission Results

We report the key summary results for the period October 2008-March 2009. The results are only a subset of queries that can be used to extract desired data from the emission database.

4.6.1 Summary of Flight & Emission Data

In the study period there were 413,359 domestic flights and 79,577 international flights. Domestic flights burned a total of 1048.57 kt of fuel and international flights burned a total of 1467.26 kt of fuel. Domestic flights flew a distance of 173,074 knm and international flights flew a distance of 122,229 knm.

Figure 4.4 presents a one-day snapshot of the domestic and international flights in Australian Airspace. Some of the trajectories in the figure are discontinued over the sea, because they belong to international flights. The terminal points are the flights' first or the last radar signature (waypoints) in the Flight Information Region (FIR) of Australia, the system received. Table 4.2 presents the total numbers of flights, distance flown, fuel used and emissions produced during the 6-month study period starting from October, 2008, and breaks them down into percentages for domestic and international flights.

Table 4.3 shows the averages of flight movements, fuel and emissions in one month for the study period. The standard deviations show that the number of

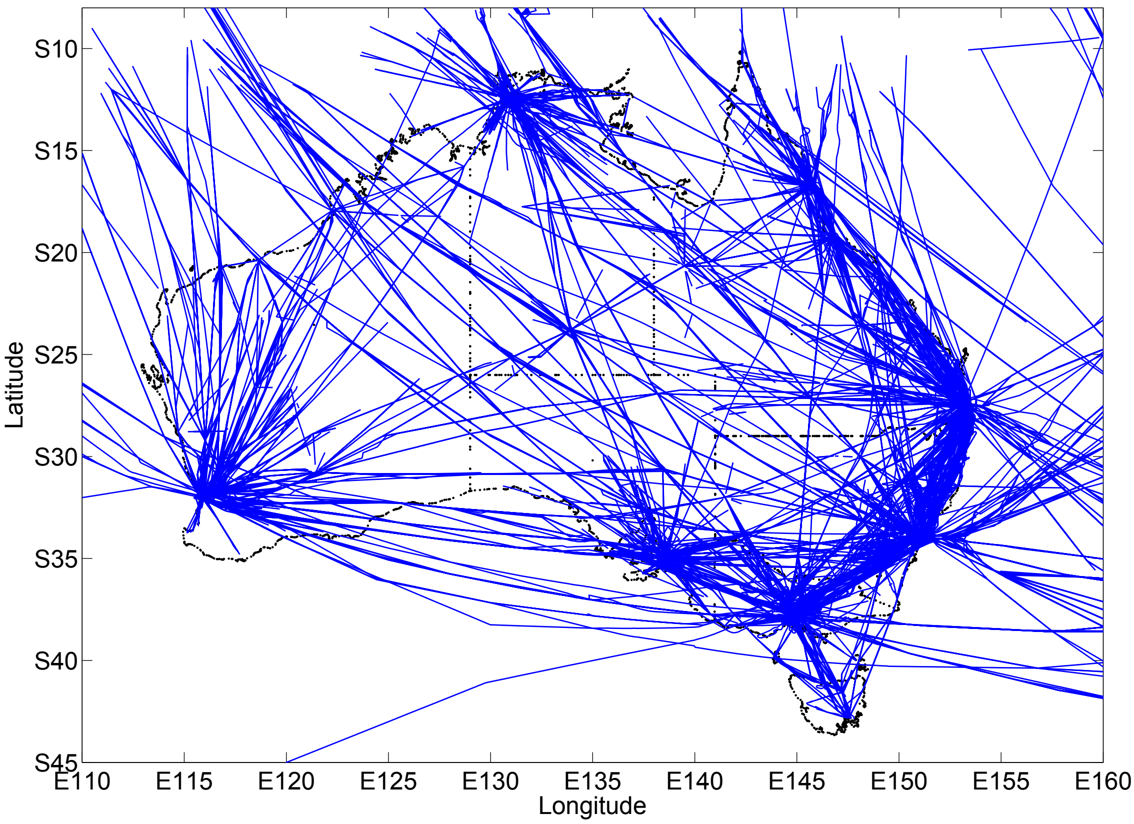


Figure 4.4: One day domestic and international air traffic in Australian National Airspace; trajectory of latitude and longitude, south (S) and East (E), respectively.

	Flights	Fuel	HC	CO	NO_x	CO_2	SO_x
Average	82,156	419.30	19.10	33.49	7.65	1321.65	0.35
Stdev	3,336	26.89	1.14	1.40	0.54	84.74	0.02

Table 4.3: Monthly averages of flight movements, fuel and emissions. Fuel and emissions are measured in kt.

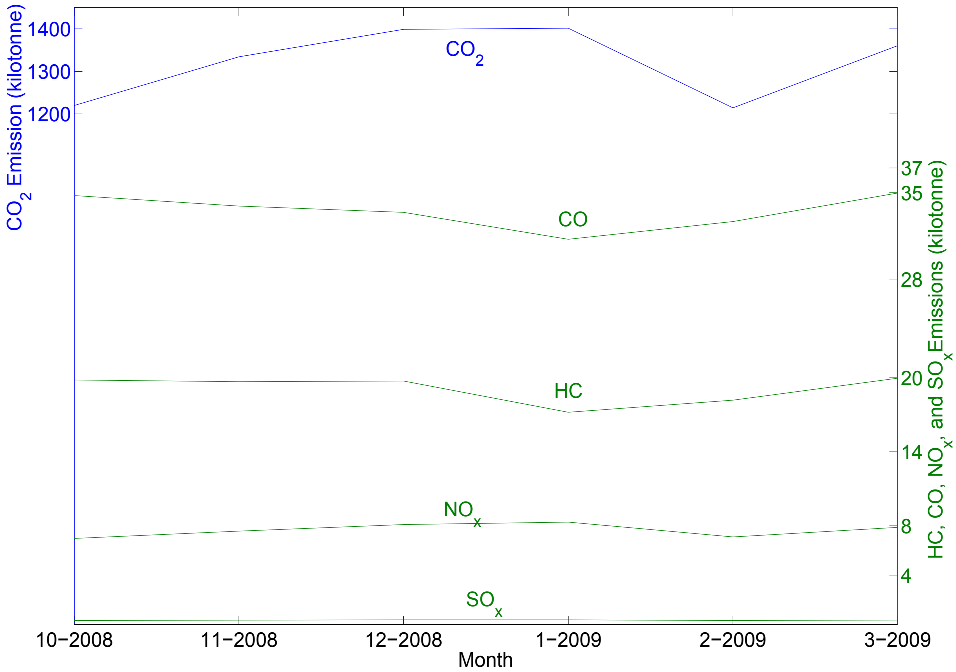


Figure 4.5: Temporal distribution of HC, CO, NO_x, CO_2, and SO_x (kt) from October 2008 to March 2009.

flights, fuel burn and emissions are not much different between the months.

Figure 4.5 presents temporal distribution in the study period. It shows a small decrease of CO_2, NO_x and SO_x in February due to reduced demand for air travel after the Christmas holiday period. We also can see that the emissions of HC, CO, NO_x and SO_x are small in comparison with CO_2 emission.

4.6.2 Fuel and Emissions of Different Aircraft Categories

Table 4.4 presents the fuel and emissions data for different aircraft categories. It can be seen that almost all HC and CO emissions are produced by turbo prop and piston aircraft, while NO_x, CO_2 and SO_x are mainly emitted by jet aircraft. 81.66%

Aircraft	Flights	Fuel	HC	CO	NO_x	CO_2	SO_x
Jet	53.08%	92.46%	5.27%	5.28%	95.74%	92.46%	92.46%
Turbo Prop	29.21%	4.83%	13.07%	31.47%	2.11%	4.83%	4.83%
Piston	17.71%	2.70%	81.66%	63.25%	2.15%	2.70%	2.70%

Table 4.4: Percentages of number of Flights, Fuel burn & Emissions based on aircraft categories

of HC and 63.25% of CO are emitted by piston flights, 13.07% of HC and 31.47% of CO by turbo prop flights, while 95.74% of NO_x, 92.46% of CO_2, and 92.46% of SO_x are produced by jet flights.

Although the total distance traveled by international flights is less than that by domestic flights, the fuel burn by international flights is more than that by domestic flights. This is because 92.51% of international aircraft types are jet, only 4.27% and 3.23% are turbo prop and piston respectively. In contrast, in domestic aircraft, jet accounts for 45.50%, followed by turbo prop 34.01%, and piston 20.50%. As a result of this aircraft distribution, most HC and CO is emitted by domestic flights, while international flights emit more NO_x, CO_2, and SO_x than domestic ones.

In general, CO and HC emissions are produced when engines are at low thrust settings (e.g. Idle, Approach etc); thus in turbo props and piston driven aircraft these emissions are high, as they are mostly short distance flights. The majority of fuel is consumed at cruise flight level of altitude, which for long and medium distance flights can be up to 70% of flight time. Thus for jet aircraft, CO_2 emissions is higher as it is directly proportional to fuel burn. NO_x and SO_x also follow the trend in fuel consumption in jet aircraft.

In Table 4.5, the fuel and emissions of each aircraft type are calculated as the average of all flights of the aircraft type for first 14 days of a month, 2008. All the aircraft in the table are jet aircraft. From the table, we can see that there is a difference in fuel consumption and emission between different aircraft types. Some aircraft/engine types consume fuel and produce emissions twice of others.

Aircraft	Fuel	HC	CO	NO_x	CO_2	SO_x
AC1	2876.66	1.45	21.75	40.40	9067.24	2.42
AC2	5240.87	2.36	13.03	144.58	16519.21	4.40
AC3	2710.29	2.40	17.05	31.19	8542.83	2.28
AC4	3016.98	2.14	12.07	34.75	9509.52	2.53
AC5	5749.47	2.34	11.61	239.74	18122.32	4.83
AC6	9208.64	31.51	65.39	219.92	29025.64	7.74
AC7	5762.91	2.37	13.33	225.81	18164.70	4.84
Average	4937.97	6.37	22.03	133.77	15564.50	4.15
Stdev	2329.92	11.09	19.45	96.85	7343.91	1.96

Table 4.5: Fuel burn & Emissions in kg for 7 representative jet aircraft; Fuel and emissions are measured in kilograms

4.6.3 Emissions distribution in different flight phases

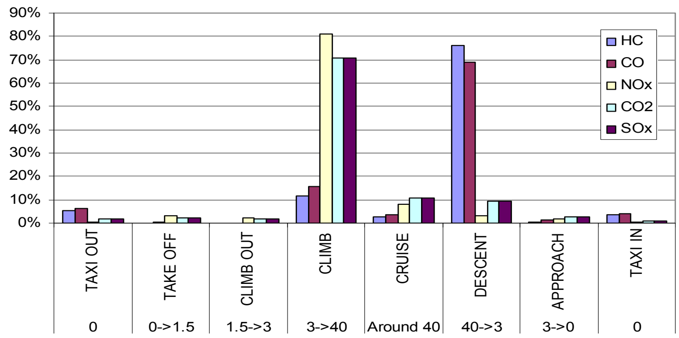
Figure 4.6 presents the aviation emissions distribution in different flight phases: Figures 4.6(a), 4.6(b), 4.6(c) present the emission distribution in different phases for a short, a medium and a long distance flight respectively, and Figure 4.6(d) shows the emission phase distribution for all the flights in the study period. The reason for presenting the altitude range of cruise phase as in the figures is that the cruise phase of a flight is determined from the top of climb to the top of descent, and the cruise altitude fluctuates slightly around a certain value. From the figures, we can see that NO_x, CO_2 and SO_x in climb phase are much higher than in the other phases in the short distance flight, while in the medium distance flight, CO_2 and SO_x are highest in cruise phase following by climb phase; NO_x in cruise phase is not much lower than in climb phase. In both short and medium distance flights, HC and CO are highest in descent phase. For the long distance flight, all emissions are mostly produced in cruise phase due to the long distance of this flight phase. Overall, Figure 4.6(d) shows that the cruise phase of flight consumes the most fuel and generates the most pollutants (more than 60%). The descent and climb phases come next, and the remaining phases account for a small amount of the total emissions. Although emissions from the descent and climb phases are much less than from the cruise phase, they are highly concentrated around airports, so the impact of these emissions is considerable. During cruise the rate of emission of HC is low. However the cruise phase accounts for so much more of a flight than other phases that the percent of HC emitted in the cruise phase is still high compared to other phases. The same applies to emissions of CO.

4.6.4 Result Comparison

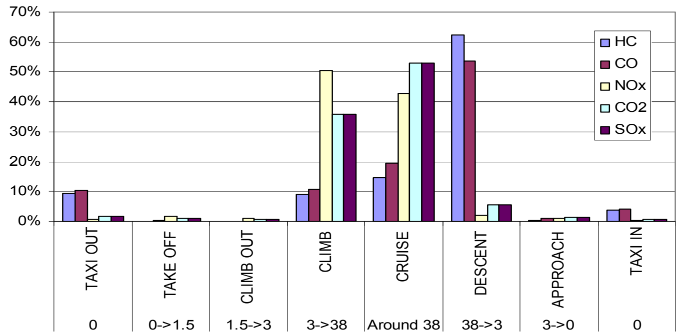
In Table 4.6, we compare the fuel burn and emission in one year for domestic and international flights in Australia from our study with SAGE in 2004 and NGGIC in 2007. In "This Study 2008", The values of variables such as "Flights", "Distance",... in one year for domestic and international flights in Australia are extrapolated from those in the 6 month study period in Table 4.2. In SAGE 2004, the number of flights ("Flights") is estimated from the total fuel burn ("Fuel") and fuel per flight ("Fuel/F"); total distance flights travelled ("Distance") is estimated from fuel per distance ("Fuel/Dis") and the total fuel burn ("Fuel").

In the table 4.6,"Flights" is the number of flights; "Distance" is the total distance flights travelled, measured in knm; "Distance/F" is the knm per flight; "Fuel" ("CO_2" or "NO_x") is the total fuel burn ("CO_2" or "NO_x") measured in kt; "Fuel/F" ("CO_2/F" or "NO_x/F") is the kt of fuel ("CO_2" or "NO_x") per flight; "Fuel/Dis" ("CO_2/Dis" or "NO_x/Dis") is the fuel ("CO_2" or "NO_x") per 1 knm; These variables are calculated for domestic ("D") flights and international ("I") flights separately; "Value" is the value of variables; "%Diff" is the percentage difference of variables between our study with other study.

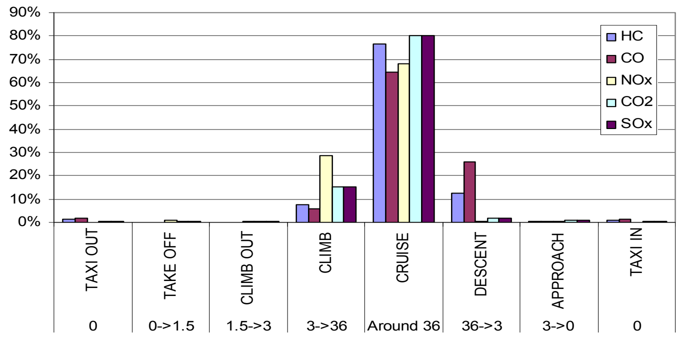
From Table 4.6, we can see that the number of flights in "This study 2008" is higher than in the two other studies. This is because air traffic grows by 4% per year. Further the definition of international flights between our study and SAGE is different. In SAGE, international flights are flights with departure airport in Australia, and arrival airport in another country, while in our study international flights are all flights whose departure airport or arrival airport is in another country. Therefore, the number of international flights in our study is much higher than in SAGE. However, the distance travelled by international flights in our study is



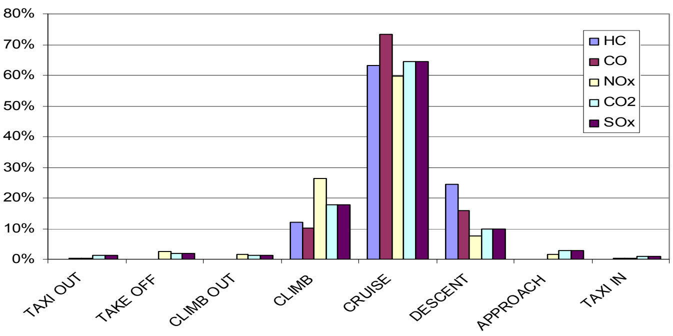
(a) Emission phase distribution of a short distance flight.



(b) Emission phase distribution of a medium distance flight.



(c) Emission phase distribution of a long distance flight.



(d) Emission phase distribution of all flights in the study period.

Figure 4.6: Emissions distribution in different flight phase. The numbers below phases in (a), (b), (c) are the altitude range of each phase, measured in kft.

		This study 2008	SAGE 2004		NGGIC 2007	
		Value	Value	%Diff	Value	%Diff
Flights	D	826718	454248	82.00%	NA	NA
	I	159154	49600	220.88%	NA	NA
Distance	D	346148	194945	77.56%	NA	NA
	I	244458	160179	52.62%	NA	NA
Distance/F	D	0.4187	0.4292	-2.44%	NA	NA
	I	1.5360	3.2294	-52.44%	NA	NA
Fuel	D	2097.13	1390	50.87%	NA	NA
	I	2934.52	2480	18.33%	NA	NA
Fuel/F	D	0.0025	0.00306	-17.10%	NA	NA
	I	0.0184	0.05	-63.12%	NA	NA
Fuel/Dis	D	0.0061	0.0071	-15.03%	NA	NA
	I	0.0120	0.0155	-22.47%	NA	NA
CO_2	D	6610.16	NA	NA	5291.94	19.94%
	I	9249.62	NA	NA	NA	NA
CO_2/F	D	0.0080	NA	NA	NA	NA
	I	0.0581	NA	NA	NA	NA
CO_2/Dis	D	0.0191	NA	NA	NA	NA
	I	0.0378	NA	NA	NA	NA
NO_x	D	29.73	19.9	49.40%	NA	NA
	I	62.12	39.9	55.69%	NA	NA
NO_x/F	D	0.000036	0.000044	-18.45%	NA	NA
	I	0.000390	0.000797	-51.03%	NA	NA
NO_x/Dis	D	0.000086	0.000103	-16.44%	NA	NA
	I	0.000254	0.000256	-0.57%	NA	NA

Table 4.6: Result comparison with other methodologies.

not much higher than in SAGE in comparison with the difference in the number of flights. The reason is that our study includes more international flights with short distance. The higher "Flights" in our study results in higher "Distance", "Fuel", "CO_2" and "NO_x" than in the other study. Another reason is that NGGIC considers the fuel consumption as the same in every segment in the whole cruise phase [131]. However, the fuel consumption is not constant in different segments in the cruise phase. Therefore, total emissions summed in every segment are likely to be different from the total emission estimated by using the fuel consumption in the whole cruise phase. This model also does not take into account international flights and other phases of flight, for instance taxi-out, climb, climb-out, approach, landing and taxi-in. As a result, the amount of aviation emissions calculated by the NGGIC methodology are likely to be an underestimate and cannot provide required insight into emissions, needed for a deeper analysis.

"Distance/F" in this study is less than in SAGE for domestic flights and especially for international flights. For international flights, the reason is the difference in the definition of international flights which results in more short-distance international flights in our study. For domestic flights, there may be more short distance flights in Australia in 2008 than in 2004. This leads to "Fuel/F", "Fuel/Dis", "CO_2/F", "CO_2/Dis", "NO_x/F", and "NO_x/Dis" in our study being less than in SAGE.

4.7 Geographical Distribution of Emission

In this section, we present the spatial analysis of emissions in Australian Airspace with cumulative emission values on a planar and a vertical display. Here, we only show the distributions of CO and CO_2 emissions, because the distribution of HC is similar to CO, and the distributions of SO_x and NO_x are similar to CO_2.

4.7.1 Spatial distribution of aviation emission

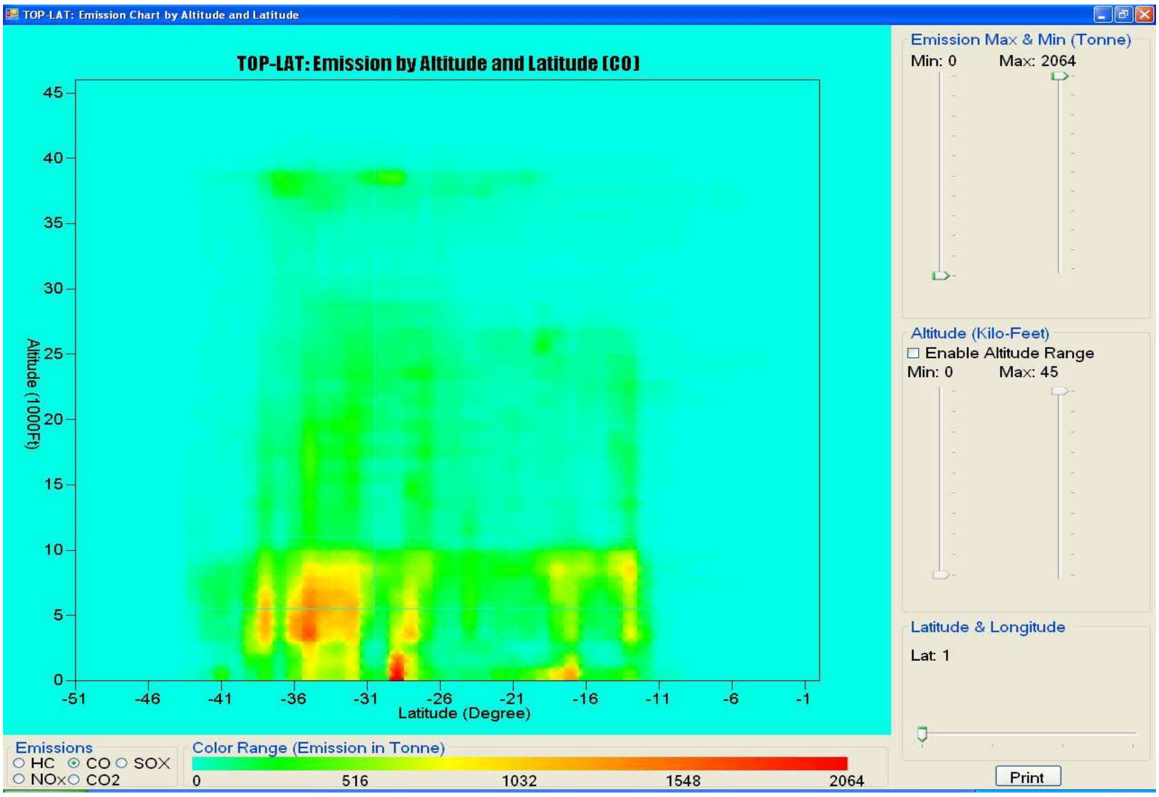
Figures 4.7 and 4.8 show that the areas around Sydney, Melbourne, Brisbane, Perth, Darwin, Cairns, and Adelaide are most impacted by aviation emissions. The emission distribution is concentrated along the South coast line because flight paths are mainly concentrated there.

Figure 4.7 shows HC and CO emissions are mainly between S31° and S36°, and below 10000 ft. They are concentrated around Brisbane, Cairns, Adelaide, Perth, Sydney, and Melbourne. HC and CO are emitted due to incomplete combustion process of jet engines at low thrust level, which explains the high concentration of CO and HC during descent and approach phases of flight. From Figure 4.8 we can see that NO_x, CO_2, and SO_x emissions affect both the local air quality and the upper airspace. Sydney, Melbourne and Brisbane airports shows high density of NO_x, CO_2, SO_x emissions. Besides that, the NO_x, CO_2, and SO_x distributions are mainly from S31° to S36°, and from 35,000 to 40,000 ft.

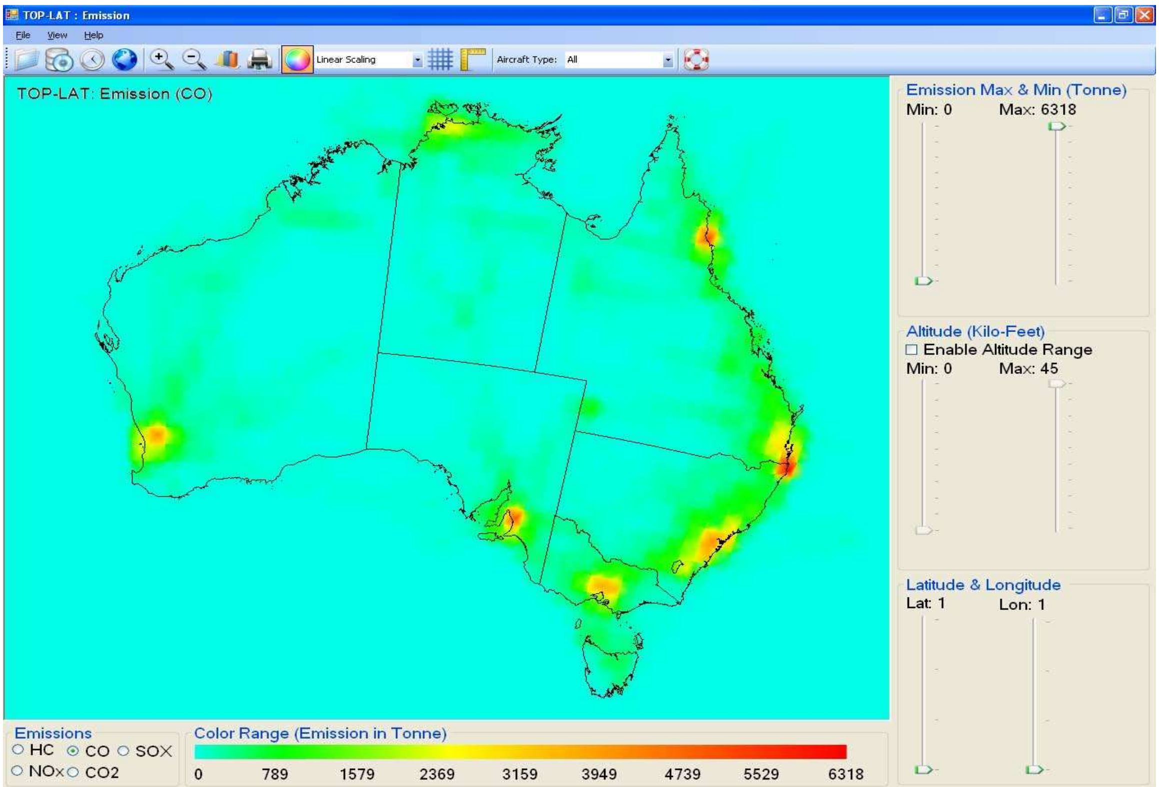
4.7.2 Distribution of Aviation Emissions by Altitude

Figure 4.9 shows that most emission of NO_x, CO_2 and SO_x is in high altitude, while emission of HC and CO is mainly in low altitude. About 50% of NO_x, CO_2 and SO_x are between the altitude band 35,000–40,000 ft and 20% between the altitude band 30,000–35,000 ft. Around 30% of HC and CO emissions are between 5,000-10,000 ft and 0–5,000 ft.

The reason for this is most jet flights, which emit a lot of NO_x, CO_2 and SO_x, cruise between the altitude band 30,000–40,000 ft, and the number of jet flights is much more than that of the other flights. On the contrary, piston aircraft, which produce a high amount of HC and CO, usually fly in low cruising altitude. Many piston flights have cruising altitude equal or less than 10,000 ft. Therefore, the percentage of HC and CO is large in altitude bands 0–5,000 ft and 5,000-10,000 ft. As a result the percentages of HC and CO in the other altitude bands are lower

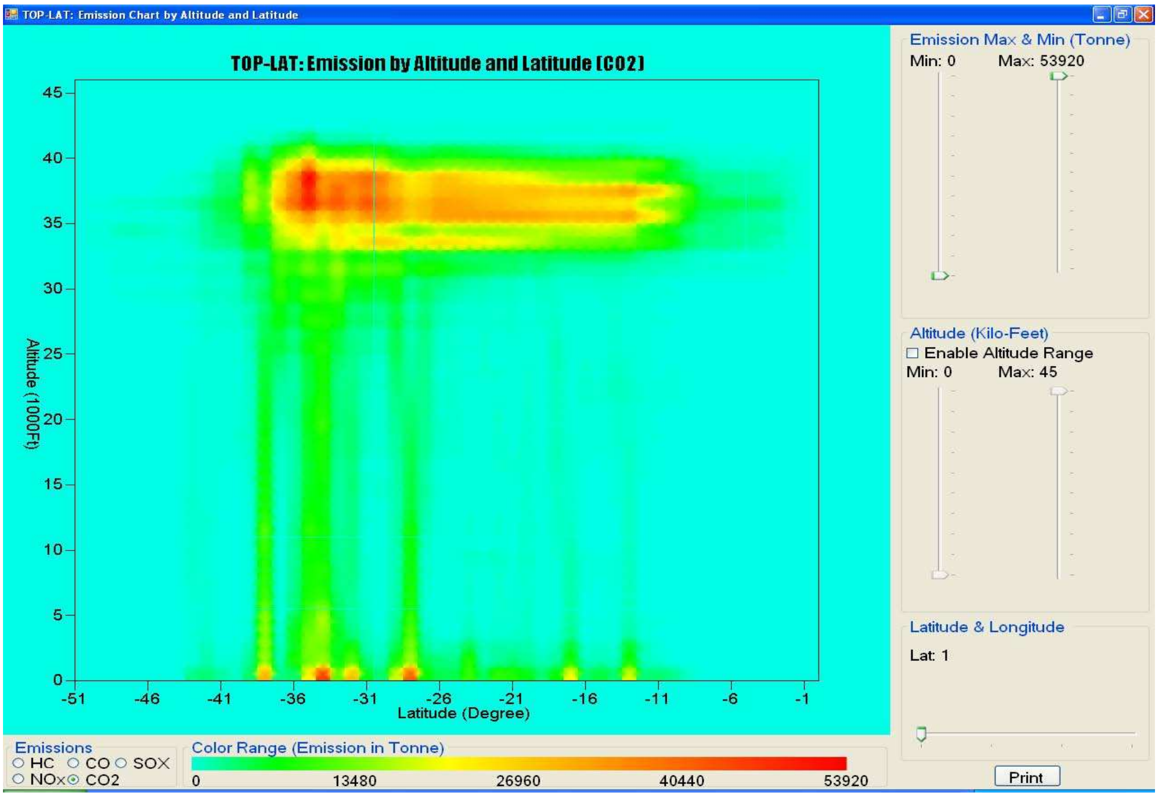


(a) CO Emission by altitude and latitude aggregated longitudinally.

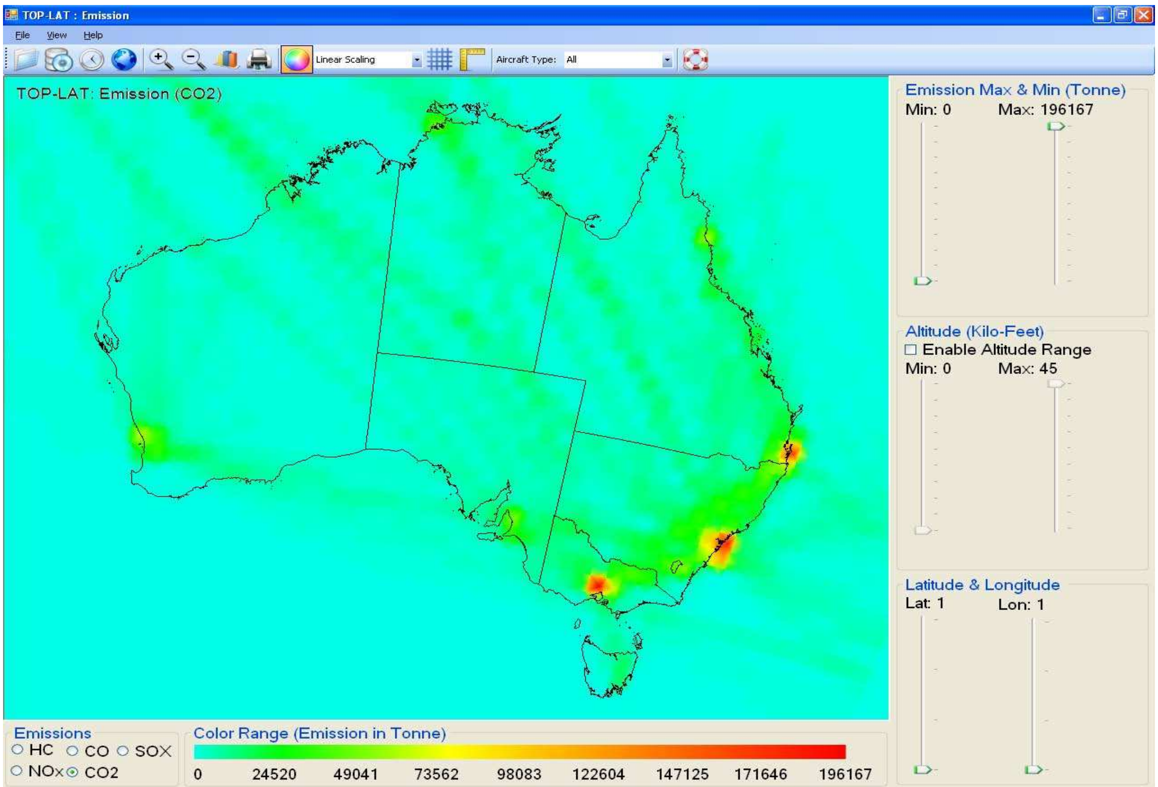


(b) CO Emission by longitude and latitude aggregated from ground level to 45 kilo-feet.

Figure 4.7: CO spatial emission distribution in the study period.



(a) CO_2 Emission by altitude and latitude aggregated longitudinally.



(b) CO_2 Emission by longitude and latitude aggregated from ground level to 45 kilo-feet.

Figure 4.8: CO_2 spatial emission distribution in the study period.

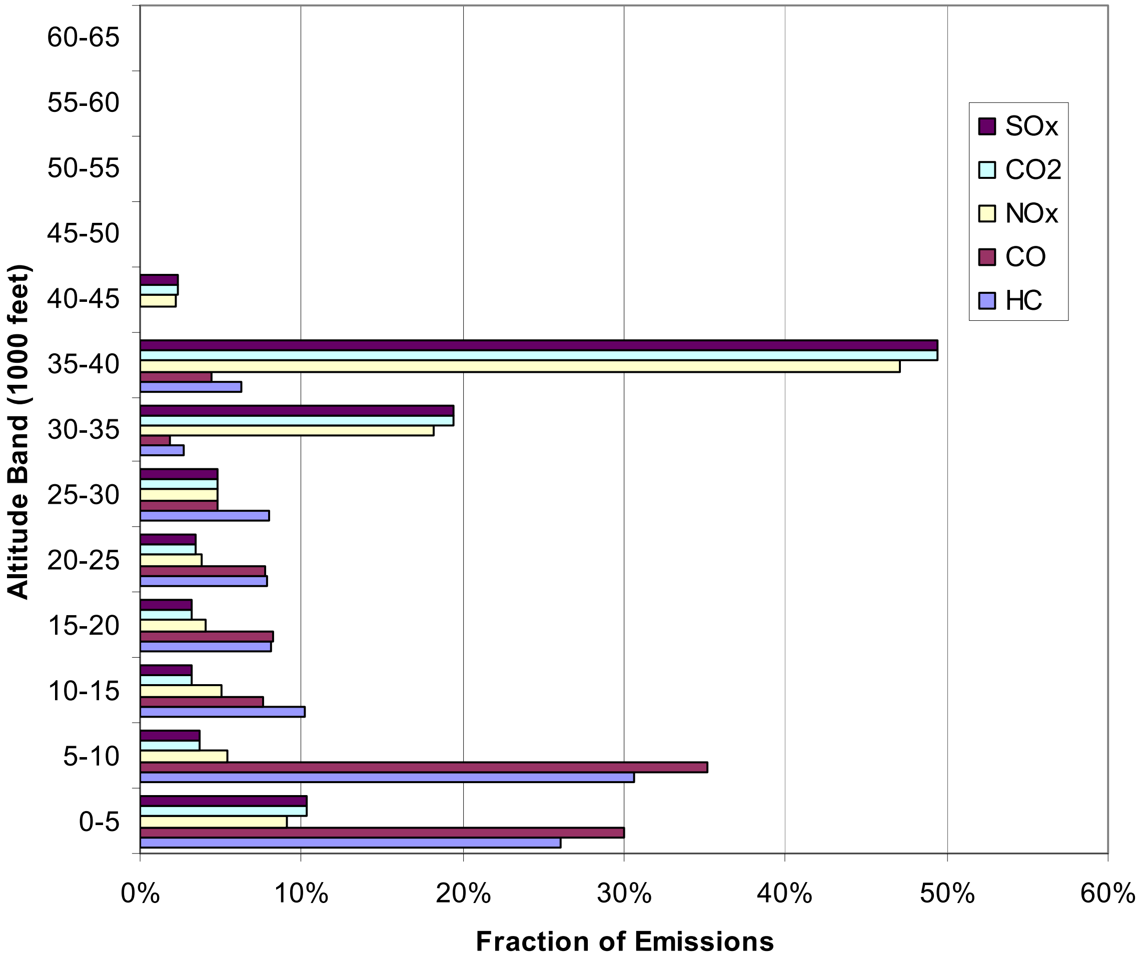


Figure 4.9: Distribution of aviation emissions by altitude in the study period.

than the two altitude bands. In addition, flights with cruising altitude higher than 10,000 ft produce a lot of HC and CO while they are at low altitude.

Because there are few flights with cruising altitude between 40,000 and 45,000 ft (high altitude band), not only the percentages of HC and CO in this band are low but the other emissions as well.

4.8 Discussion

From the aviation emissions data, we found that the amount of emitted CO_2 at several cells is much higher than the others. Such cells are identified in Table 4.7, where "long (deg)" is the longitude of a cell in degree; "lat (deg)" is the latitude of

Long (deg)	Lat (deg)	CO_2 (kt)	Airports
E144	S38	196.17	Melbourne, Point Cook, Yarra Bank, Penfield, Essendon
E151	S34	177.46	Sydney
E153	S28	162.87	Brisbane, Archerfield, Dunwich
E150	S34	97.6	Bankstown, Westmead, Richmond, Glenbrook, Katoomba
E115	S32	63.79	Perth, Gingin

Table 4.7: 5 grid cells with largest amount of CO_2 emitted in troposphere layer.

a cell in degree; "CO_2 (kt)" is the kilotonnes of CO_2 emitted at a cell in troposphere layer in the 6-month study period; "Airports" is the list of airports at a cell, the amount of CO_2 of one cell is summed from ground level to an altitude of 14 km. This altitude is used, because the depth of the troposphere in Australia varies latitudinally, seasonally, and daily. For example, it is about 16 km above Australia at year-end, and between 12–16 km at midyear, being lower at the higher latitudes [177, p. 23]. Therefore we adopt a mean depth of 14 km for the troposphere to calculate CO_2 from the ground to this altitude. If the air traffic growth of 4% per year [168] and other emissions sources are taken into account, the temperature around these cells may increase faster than in other regions, potentially with significant impact on the local people's health.

Because the main species for ozone change in aviation emissions is NO_x, we try to find some cells where flights emit NO_x much more than at others. Furthermore, since most aviation emissions in Australia occur below 45,000 ft (about 13.7 km) (see Figure 4.9), we only consider cells in the upper troposphere in Australia between 10 km and 14 km (about 32,000 ft to 46,000 ft), where ozone is one important greenhouse gas. This means the NO_x emission of aviation in Australia does not impact directly on the ozone layer in the stratosphere, where ozone absorbs the sun's high frequency ultraviolet light. Understanding how ozone varies will reveal how this constituent buffers the temperature increase. In Table 4.8, we found some cells between 10 km and 14 km with much more NO_x emitted than the other cells at the same altitude level. In Table 4.8 "long (deg)" is the longitude of a cell in degree; "lat (deg)" is the latitude of a cell in degree; "NO_x (kt)" is the kilotonnes of NO_x emitted at a cell between 10 km and 14 km altitude in the 6-month study period. In the future, we will consider other factors such as wind effect and chemical

Long (deg)	Lat (deg)	NO_x (kt)
E148	S34	0.14
E119	S14	0.13
E118	S13	0.13
E146	S37	0.13
E149	S33	0.12
E120	S15	0.12
E148	S36	0.12
E150	S32	0.12
E152	S30	0.11
E121	S16	0.10

Table 4.8: 10 grid cells with largest amount of NO_x emitted in upper troposphere layer

interactions which affect the concentration and distribution of emissions to estimate the impact of aviation on the environments as a whole and at these cells we found above. That may provide a better view of aviation emissions impact. The system's output of gridded emissions with high temporal resolution may also be incorporated with general emission processing systems or specific chemical-transport models to create a general picture about emissions in Australia. This will help environmental specialists to analyse the impact of emissions as well as the air quality in Australia.

4.9 Limitations and Assumptions

The first limitation in this study arises from lack of availability of high fidelity emission & fuel flow data. An aviation engine's fuel consumption and emissions at higher altitudes is commercially sensitive. The second limitation arises from some of the assumptions underlying the study. In this section, the most significant assumptions are summarized and discussed.

Flight with a standard payload and no fuel tankering: Payload affects the fuel flow and hence the emission computation. However, data about payloads or actual fuel amounts carried by the individual flights are not available from ATC data. Thus we have assumed an average payload and do not account for fuel tankering.

No consideration of delays and holdings: Ground based (taxi) delays may be approximated by the use of aircraft-specific times-in-modes, whereas airborne delays, such as holding, are more difficult to consider if not captured in trajectory data.

Simplified routing and trajectory modeling: Flight routing does not account for different mixing heights at different airports and are assumed fixed at 3000 ft. Further noise abatement procedures are not considered, simplifying the landing and take off cycle. No aircraft and engine deterioration: Since such data is difficult to obtain and model, the effects due to deterioration and engine aging are excluded in our study.

Military, helicopters, and recreational flights are not included: Apart from the fact that military flight data is classified and data on engine and aircraft type is not available, military flights mostly operate in special use airspace and use upper airspace (above 41,000 ft). Emission modeling for helicopters is not yet developed and they constitute a very small number of operations. Data on the engines of recreational aircraft is also not available.

A series of parametric studies was conducted to evaluate the effects of wind, temperature, payload, tankering, and cargo on the calculated fuel use [18]. These studies show that the main uncertainty is in the assumption about fuel tankering, in which the actual fuel burn is 8% more than the fuel estimated with no fuel tankering. Next in uncertainty are assumptions about passengers load factor and wind, where the uncertainty is about 2.5%. The uncertainty caused by assumption about standard temperature is less than 1%. Therefore, the overall uncertainty in our assumptions and limitations is small (likely less than 10%).

Another limitation of the inventory is that it does not take into account that emissions can be moved around by wind after they have been emitted. Particle emissions are also not available in this inventory currently. These are areas for possible future work.

4.10 Recommendations

In terms of air traffic management, several options exist to slow the increase of CO_2 concentration at polluted cells. Current major airports might be replaced with several other airports in different areas. Passengers will travel to the new airports by other traffic vehicles which consume less fuel. As a result, the emissions will be distributed more evenly to different areas. While the development of multi-airport systems involves and is influenced by a wide range of factors [130], empirically verifiable concerns about air transport noise and atmospheric pollution are certainly one of them [73].

Rerouting flights may help to reduce the emission concentration at congested waypoints and flight paths. Because the depth of the troposphere in Australia varies regionally, seasonally, and even daily, using flight levels suitable to the regions and seasons might reduce the impact of NO_x emissions on the ozone layer.

Flights can use direct routes to save time and fuel burn. The Direct-To Controller Tool identifies that aircraft can save at least one minute of flying time by flying direct to a down-stream fix along its route of flight [188]. Our system (TOP-LAT) also provides User-Preferred Trajectories tool for optimising flight trajectories in terms of time and distance travelled, fuel burn, emissions produced, and comfort for passengers.

Efficient departure and arrival planning can also help to reduce the time aircraft are in runways. As a result, aviation emissions will become less in runways. For example, the Departure Enhanced Planning And Runway/Taxiway-Assignment System (DEPARTS) developed by the MITRE Corporation's Center for Advanced Aviation System Development can generate optimised recommendations on runway assignment, departure sequencing and departure fix loading for the air traffic control tower to reduce taxi-out times [101].

In terms of aviation technology, inventing new engine types which consume less fuel, or another energy which does not produce emissions, would also be a solution.

Such decisions could not be made without careful consideration of all their effects. The emission inventory presented here will be an essential element in that process.

4.11 Chapter Summary

In this chapter we present an aviation emission inventory for Australian Airspace, and some discussion based on a subset of data derived from the inventory. This is the first attempt to develop an emission inventory using real time trajectory and flight plan data. The emission database is prepared by TOP-LAT. This system is online and processes air traffic data to compute aviation emissions in real time. The emission inventory is an ongoing process, making use of real time data as it arrives. The inventory is in the form of a 4D database which provides resolution of 1° x 1° x 1,000 ft, for temporal and spatial emission analysis.

The aviation emissions are calculated based on the actual 4D trajectories which are mixed of actual flight routes and radar data. This can help provide more accurate emission data in comparison with other emissions inventories which only use flight plans and assume a great circle route between origin and destination airports. Some information of a flight is not given sometimes. They are also estimated to complete information about the flight. This further improves the accuracy of emission computation.

The specific emission inventory reported here covers a period of six months from October 2008. The study processed 492,936 flights for the given period, which burned 2,515.83 kt of fuel and produced 114.59 kt of HC, 200.95 kt of CO, 45.92 kt of NO_x, 7,929.89 kt of CO_2, and 2.11 kt of SO_x.

For emission computation, the BEM2 methodology along with ICAO engine emission data bank is used. Fuel flow is computed using aircraft specific aerodynamic and thrust model, with weight correction and accounting for atmospheric data.

The emission results show that domestic flights produce much more HC and

CO than international flights, conversely international flights emit more NO_x, CO_2, and SO_x than domestic ones. The results also show that HC and CO emissions are mostly produced by turbo prop and piston, while NO_x, CO_2 and SO_x are emitted by jet aircraft. 81.66% of HC and 63.25% of CO are emitted by piston flights, and 13.07% of HC and 31.47% of CO by turbo prop flights, while 95.74% of NO_x, 92.46% of CO_2, and 92.46% of SO_x are produced by jet flights.

The study found that the major international airports such as Sydney, Melbourne and Brisbane airports show high density of aviation emissions. In addition, the coast line is much more impacted by aviation emission than the other regions.

All the emission distributions are mainly between S31° and S36° latitude.

Most emission of NO_x, CO_2 and SO_x is in high altitude (the cruising altitude of most jet flights), while emission of HC and CO is mainly in low altitude (the cruising altitude of most piston flights). About 50% of NO_x, CO_2 and SO_x are between the altitude band 35,000–40,000 ft and 20% between the altitude band 30,000–35,000 ft. Around 30% of HC and CO emissions are between 5,000-10000 ft and 0–5000 ft.

The study found that several cells around major airports such as Melbourne, Sydney and Brisbane have CO_2 emitted by air traffic much more than the other regions, and some cells in the upper troposphere with much more aviation emissions of NO_x than the others. These areas may be environmentally impacted more than the other areas.

It is expected that with the availability of a real time aviation emission database for Australian Airspace, environmental analysts and aviation experts will have an indispensable source of information to support timely decision-making regarding expansion of runways, building new airports, applying route charges based on environmentally congested airways, and restructuring air traffic flow to achieve sustainable air traffic growth.

In the future when we have more information about the assumptions that we use in this study, we will update the system to compute fuel burn and emissions more accurately. We will also analyse the impact of aviation emissions on the environment in Australia such as the effect on the ozone layer, and greenhouse effects, in which wind effect and chemical interactions will be taken into account.

Chapter 5

3-D User Preferred Routes using Black and White Box Approaches

This chapter is partially based on following publications:

1. V. V. Pham, L. Bui, S. Alam, C. Lokan and H. A. Abbass. A Pittsburgh Multi-Objective Classifier for User Preferred Trajectories and Flight Navigation, IEEE Congress on Evolutionary Computation, ISBN: 978-1-4244-6909-3, 2010.

5.1 Overview

In this chapter, both black and white box approaches are designed to find 3-D (latitude, longitude, altitude) user preferred routes for flights in both problems GRP and GAP (which are formulated in Section 3.2). A black box is an algorithm that does not provide a reason for the choice of a solution (route), while a white box provides the cause of choosing a solution (route).

Genetic algorithms (GA) are designed as black box approaches to find 3-D UPR routes. The output of this heuristic method is a route description (a chromosome). We can understand the outcome of GA, but not the underlying cause. Therefore the algorithm works as a black box. GA is run for each flight to find its preferred route. This route description is simulated to get the 4-D trajectory. The 3-D route is obtained by removing the time dimension from the 4-D trajectory.

Though the simulation can be applied to obtain 4-D trajectory, we consider the problem as 3-D user preferred routing. The reason is that only the 3 dimensions (latitude, longitude, altitude) of the trajectory are optimised, while the speed of the aircraft in the 4-D trajectory (which is used to determine the time dimension of the trajectory) is not optimised; it is calculated by linear approximation from BADA database as presented in Section 3.6.1.

Learning Classifier Systems are designed as white box approaches. This heuristic method outputs the classifier as a decision table of ordered rules in plain language. This makes solutions understandable to users and also points out the cause of every movement of the aircraft. The Learning Classifier Systems find the best classifier to optimally navigate all flights simultaneously.

The conflicts between the user preferred routes, which are provided by GA and LCS, are then detected and resolved by the same model presented in Section 3.8.

The experiments are implemented with different flight and weather scenarios for both black and white box approaches. The hypothesis is that in addition to the advantage of being transparent to users, the white box approaches can achieve solutions that are as competitive as the black box approaches.

The remainder of the chapter is organized as follows. Section 5.2 presents the design of Genetic Algorithms applied to both black and white box approaches, as well as for the two UPR problems: GRP, and GAP. A summary of the experiments is also presented in this section. Section 5.3 presents the black box approach, where Genetic Algorithms are designed to solve the two UPR problems. Section 5.4 presents the white box approach, where Learning Classifier Systems are designed to find UPR routes for the two problems. Section 5.5 compares the two approaches in terms of running time and how good the UPR routes provided by these algorithms are, and discusses the departure time adjustments of UPR flights for conflict resolution. Conclusions are presented in the last section.

5.2 Genetic Algorithm for Black and White box Approaches

Both black and white box approaches use a Genetic Algorithm to evolve a population of individuals. An individual in the black box approach represents a set of commands to design a route, while an individual in the white box approach represents a classifier. Algorithm 4 presents Genetic Algorithm for both approaches used in this chapter. Here we use Genetic Algorithm (GA) rather than other heuristic methods, as GA is a population-based method so it has flexibility in choosing a solution, instead of providing only one solution. Other techniques can record multiple solutions. However the same algorithm may need to be used many times. In addition, as there are a number solutions being processed simultaneously, it is expected that GA is able to find a global solution. Furthermore, a genetic algorithm is a stochastic search method, so it does not require the fitness function to be smooth, while a traditional search method that follows the path of least resistance does.

This algorithm evolves a population of *popsize* individuals through *ngen* generations. The genetic operators (selection, crossover, mutation) are used to reproduce a population. These operators are standard; my aim is not to develop innovative operators, but to apply GA in an innovative way. Particularly, binary tournament selection is used. The standard one point cross-over with the crossover probability of 1 is used. The mutation operator is implemented differently for different individual representations, depending the structure of an element in the individual. The mutation probability is investigated for use. The best individual is maintained in the population by replacing a worse individual, which is chosen randomly in the child population, with the best individual in the parent population. The population size is 100, the number of generation is 100.

The generation of flight data, flight scenarios, and weather scenarios generations was presented in Sections 3.9.1, 3.9.2, and 3.9.3. In this algorithm, these data are loaded as input for the algorithm. The flights can be given routes or cruising altitudes, which correspond to the two problems GRP, and GAP.

Algorithm 4 Genetic Algorithm

```
Load wind and bad weather data.
Load flights.
popsize is the number of individuals in a population.
ngen is the number of generations in the evolution process.
Initialize parent population of popsize individuals.
Evaluate the parent population.
for (i=2; i≤ngen; i++) do
    Select every two individuals in the parent population, cross-over the two to create two new individuals and
    add them to a new population called the child population. This is repeated until the child population has
    popsize individuals.
    Mutate the child population.
    Evaluate the child population.
    Choose a random individual in the child population.
    if the chosen individual is worst than the best individual of the parent population then
        Replace the chosen individual in the child population with the best individual of the parent population.
    end if
    Copy all the individuals from the child population to the parent population.
end for
Return the best individual of the parent population.
```

The differences in using Algorithm 4 for the black and white box approaches and the two problems are the representation and the evaluation of an individual. These are presented in the following sections for each approach and problem.

5.3 3-D User Preferred Routes using Genetic Algorithms

In this section, GA is designed for finding the 3-D user preferred route for a flight. We propose a new representation and evaluation for a chromosome. We also design the algorithm to evaluate a chromosome. A 4-D trajectory is achieved through simulation using the 3-D route description with the given departure time and aircraft performance from BADA database. The 3-D route is then obtained by removing the time-dimension from the 4-D trajectory. We experiment with mutation probability to choose the best mutation probability. The quality of solutions is evaluated using weighted sum of time, and discomfort.

5.3.1 3-D User Preferred Routes using Genetic Algorithms for flights given 2-D routes

In this section, a Genetic Algorithm is used to find the 3-D user preferred route for the GRP problem, where a flight is given 2-D (latitude, longitude) route and the flight can change its altitude during cruise phase.

The experiment design is presented in Section 5.3.1.1. In this section we discuss and propose a suitable representation for a chromosome, representing a flight path. Genetic operators are presented next. Then we present the algorithm to generate the 4-D trajectory for a flight, using a chromosome. The evaluation of a chromosome is then presented. Finally, the mutation probability is investigated to find the best mutation probability. The objective of experiment is to explore the behavior of GA and comparing it with local and random search.

Some results are presented in Section 5.3.1.2.

5.3.1.1 Experiment design

a) **Representation**

A flight path can be represented as a series of points. Each point is defined by 3 coordinates (latitude, longitude, altitude). This representation leads to some chromosomes giving un-implementable routes. For example, the aircraft may go backward, as the next point in a series may be behind the point in the airspace in the direction from the origin to the destination, or with the given aircraft performance, the aircraft cannot perform all the changes in latitude, longitude, and altitude between the two consecutive points.

The chromosome can be reordered so points are in forward direction. However this representation destroys building blocks of the chromosome.

In this thesis we propose to navigate a flight by a series of commands (or segments). A command tells an aircraft to climb, cruise, or descend in a relative length. The relative length is then transferred to an additional climb/descent altitude, or a cruise distance that the aircraft can perform. This can also make sure the aircraft goes in the direction towards the destination.

A chromosome is represented as a series of climb, cruise and descent segments for a flight. The size of the chromosome (number of segments) for a flight depends on the great circle distance of the flight and it is equal to $\frac{GCD}{100}$, where GCD is the great circle distance of the flight in km. A segment is represented by movement type (climb, cruise, descent) *movetype* and a relative length l, which is an integer value from 1 to $MaxLen$. $MaxLen$ is the maximum length of the segment. The absolute length (the additional altitude or the additional cruise distance) of a segment is determined as follows:

- If a segment is a climb segment, the additional climb altitude is $\frac{l*MAA}{MaxLen}$ (ft), where MAA (the Maximum possible Additional Altitude in feet) is the difference between the maximum cruising altitude the aircraft can climb to and the current altitude of the aircraft.
- If a segment is a descent segment, the additional descent altitude is $-\frac{l*MAA}{MaxLen}$ (ft), where MAA is the difference between the current altitude of the aircraft and the minimum cruising altitude. If the maximum cruising altitude is higher than 10000 feet, then the minimum cruising altitude is 10000 feet. Otherwise the minimum cruising altitude is the maximum cruising altitude.
- If a segment is a cruise segment, the additional cruise distance is $l * 100$ (km).

The parameters we used are: the maximum length of a segment $MaxLen$ is 5, the distance of a default cruise segment is 100km, the number of segments in a chromosome is $\frac{GCD}{100}$, where GCD is the great circle distance between the origin and destination of a flight measured in km (this number is approximate to the number of cruise segments for a flight to fly from the origin to the destination).

Figure 5.1 shows the chromosome structure.

b) **Genetic Operators**

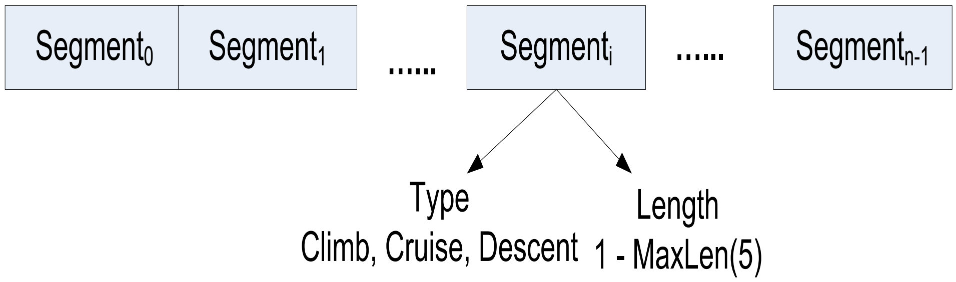


Figure 5.1: Chromosome for GA, flights are given routes

The standard genetic operators are used as presented in Section 5.2. The mutation operator goes through every segment in a chromosome. For each segment, a random number from 0 to 1 is generated. If the number is less than the mutation probability, the segment is mutated to become a new segment by changing its type of movement (e.g from a cruise to a climb) and changing the length of the segment from 1 to a maximum length $MaxLen$.

c) **Generation of the 4-D trajectory**

The algorithm to simulate the series of segments to obtain a 4-D trajectory is separated into two parts: Algorithm 5 and Algorithm 6, because of the algorithm length. The simulation follows the aircraft performance constraints, and takes wind information into account. First all the TODs at every flight level from the minimum cruising altitude to the maximum cruising altitudes are determined (as the simulation then will continuously check whether the aircraft passes the TOD at the altitude of the ending point of a segment or not). The simulation starts from the origin. The first segment tells aircraft to climb to the Top Of Climb (TOC) point. The next segments can tell aircraft to climb, cruise, or descend. If after flying a segment, the aircraft passes the TOD at the altitude of the ending point of the segment, the segment will not be added to the 4-D trajectory; instead the aircraft will cruise to the TOD at the altitude of the current point and then continuously descend to the destination. If after flying all the segments the aircraft has not reached the destination, it will cruise to the TOD at the altitude of the current point and then continuously descend to the destination. If a segment tells the aircraft to climb or descend but the current altitude of the aircraft is the maximum or minimum cruising altitude, the segment will be changed to a default cruising segment with a predefined cruise distance. Here, the simulation of the series of segments integrates the continuous descent approach, which is proved to save fuel, emissions (including CO_2 emission and noise emission) in literature [195, 116, 5].

Algorithm 5 Generation of the 4-D trajectory for a flight given 2-D route, using a chromosome - Part I

Determine TOD points at the flight levels from the minimum cruising altitude to the maximum cruising altitude (depending on the flight). In order to determine TOD at a flight level, Algorithm 3 in Section 3.7.7 is used to backward simulate the aircraft following a 2-D route to descend from a starting point given altitude to an ending point, where the ending point is the destination and the starting point is TOD whose latitude and longitude need to determine.
Determine the additional climb altitude of the first segment as presented above.
Determine the target altitude of the first segment. It is first calculated as the altitude of the origin airport plus the additional climb altitude. It is then rounded to become a flight level.
Algorithm 1 in Section 3.7.1 is used to simulate a 2-D route to climb to the target altitude, where the starting point is the origin airport. TOC is the ending point of the simulated segment.
The current point is TOC.
i=1
exit = false
while i < number of segments (number of elements in the chromosome) and not exit **do**
    if segment$_i$ is a climb (or descent) segment **then**
        Determine the additional climb (or descent) altitude as presented above.
        Determine the target altitude of the segment. It is first calculated as the altitude of the current point plus the additional climb (or descent) altitude. It is then rounded to become a flight level.
        if segment$_i$ is a climb (or descent) segment and the target altitude is the maximum (or minimum) cruising altitude **then**
            The segment becomes a default cruise segment.
        end if
    end if
    if The segment is a climb segment **then**
        Simulate a 2-D route to climb to an altitude by Algorithm 1 in Section 3.7.1.
    end if
    if The segment is a descent segment **then**
        Simulate a 2-D route to descend from a starting point to a target altitude by Algorithm 2 in Section 3.7.4.
    end if
    if The segment is a cruise segment **then**
        Determine the ending point of the segment using the current point and the distance between the current and the ending points.
        if The ending point of the simulated segment does not pass TOD at the altitude of the ending point **then**
            Algorithm presented in Section 3.7.10 with algorithm flow in Figure 3.17 is used to simulate the aircraft following a 2-D route to cruise from the current point to an ending point.
        end if
    end if
    {**while loop continues in the next page**}
    ...
end while

Algorithm 6 Generation of the 4-D trajectory for a flight given 2-D route, using a chromosome - Part II

while i < number of segments (number of elements in the chromosome) and not exit **do**
  {**while loop continues from the previous page**}
  ...
  if The ending point of the simulated segment passes TOD at the altitude of the ending point **then**
    The simulated segment is not taken into the trajectory.
    Aircraft cruises from the current point to TOD at the altitude of the current point and continuously descends to the destination. Algorithm presented in Section 3.7.10 with algorithm flow in Figure 3.17 (which is used to simulate the aircraft following a 2-D route to cruise from a starting point to an ending point) is used, where the starting point is the current point and the ending point is TOD.
    The cruise segment from the current point to TOD, and the descent segment from TOD to the destination are added into the trajectory.
    exit = true
  else
    The simulated segment is added into the trajectory.
    The current point is the ending point of the simulated segment.
  end if
  i=i+1
end while
if exit is false **then**
  Aircraft cruises from the current point to TOD at the altitude of the current point and continuously descends to the destination. The simulation as presented above.
  The cruise segment from the current point to TOD, and the descent segment from TOD to the destination are added into the trajectory.
end if

Figure 5.2 shows an example of generating the 4-D trajectory for a flight whose maximum cruising altitude is 40 kft, and the distance from the origin to the destination is 700 km. The number of elements in a chromosome is 7. The chromosome is presented in the top of the figure. A element includes two fields: segment type and length. A segment type can be Cl (Climb), Cr (Cruise), or D(Descent). A segment length can be from 1 to 5. The first six segments tell the aircraft to climb to 24 kft, cruise a distance of 100 km, climb to 33 kft, cruise a distance of 200 km, descend to 21 kft, then cruise a distance of 200 km. These climb, descent altitudes are the altitudes of the current points plus the additional climb, descent altitudes (which are calculated as in (a)). The additional distance is calculated as in (a). The last segment tells the aircraft to climb to 28 kft. However after climbing to 28 kft, the aircraft passes the TOD at 28 kft, so this

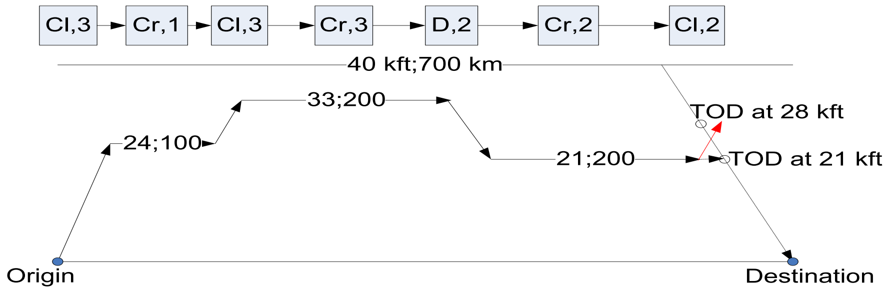


Figure 5.2: An example of simulation for chromosome representing a flight given a route

segment is not added to the 4-D trajectory, the aircraft cruises directly to TOD at the altitude of the current point at 21 kft, and then continually descends to the destination.

The 4-D trajectory is converted to the 3-D route by removing the time dimension.

d) **Evaluation**

Assume that the 4-D trajectory has n simulated segments from 0 to $n-1$. For all segments, time, discomfort and fuel are calculated. The combined objective from time and discomfort is

$$Obj = \sum_{i=0}^{n-1} u * time_i + (1-u) * BWL_i * time_i \tag{5.1}$$

where $time_i$ is time the aircraft travels through the segment i, BWL_i is the bad weather level at the first point of the segment i. The bad weather level in a segment is assumed to be constant at every point in the segment. $BWL_i * time_i$ is the discomfort of the segment i.

Fuel is not considered in the objective function. It is only considered indirectly, through time minimization and the decision to assume continuous descent.

e) **Mutation probability investigation and parameter settings**

Mutation probability investigation

We did an experiment with the following settings to find a good mutation probability for GA.

Mutation probability	0	0.1	0.2	0.3	0.4	0.5	0.6
Percentage of flights	8.5%	45.1%	31.0%	35.2%	29.6%	26.1%	27.5%

Table 5.1: Mutation probability investigation; GA for flights given routes

- Flight Scenario: The flight scenario $UA2$ is used where one third of UPR flights is allocated for each of the 3 user preferences.
- Weather Scenario: Normal Wind, Bad Weather Cells affecting 50% of UPR flights.
- Crossover probability: 1
- Mutation probability can be 0, 0.1, 0.2, 0.3, 0.4, 0.5, or 0.6

For each flight, the algorithm was run with 6 different mutation probabilities (0, 0.1, 0.2,... 0.5, or 0.6) and with 10 seeds. The average objective of the 10 best individuals on the 10 seeds for each mutation probability was determined for a flight. A mutation probability is the best for a flight if its average objective is the smallest. Table 5.1 presents the percentage of flights for which each mutation probability is the best. The total of all the flight percentages is more than 100% as there may be more than one mutation probability with the average objective being the smallest for a flight. The table shows that the mutation probability of 0.1 is the best in general. This also proves that GA with mutation probability of 0.1 works better than local search and random search, which correspond to GA with mutation probability of 0, and 0.6 respectively. The percentage of flights for which the mutation probability of 0.1 is the best is 45.1% while those for the mutation probabilities of 0 and 0.6 are only 8.5% and 27.5% respectively. Therefore we choose the mutation probability of 0.1 for the algorithm.

5.3.1.2 Results

a) **Generation of best individuals and the minimum objective through generations**

Table 5.2 shows the average number of generations at which the best individuals

User Pref.	BW0 - 0%	BW1 - 10%	BW2 - 50%	BW3 - 100%
Discomfort	NaN	34	36	37
Neutral	NaN	49	50	48
Time	49	49	49	49

Table 5.2: The average of generations where the best individual is found; GA for flights given routes.

are found. In the table "User Pref." is the user preference to a flight; "BW0 - 0%" is the weather scenario without bad weather cells; "BW1 - 10%", "BW2 - 50%", and "BW3 - 100%" are weather scenarios affecting 10%, 50%, and 100% of flights. "Discomfort", "Neutral", and "Time" are user preferences or utilities of 0.000001, 0.5, and 0.999999. The cells of "BW0 - 0%" column and "Discomfort" and "Neutral" rows are "NaN", as we chose only one flight scenario, in which all prefer to minimize time, to experiment with the weather scenario with no bad weather cells.

Figure 5.3 shows the minimum objective through generations of an example flight. The flight has the chromosome size of 20 (the average size of chromosomes). It is given a route and prefers to minimize time or discomfort. The 10 different colored traces in each sub-figure are for 10 different runs.

Figure 5.3(a) presents the minimum objective for a flight for which the the user preference is to minimize time. As the discomfort is multiplied by a very small number 0.000001, the minimum objective can be considered to present time objective. The figure shows that the objective values from 10 different seeds in the final generation are only about 1 minute different from each other. The minimum objective ranges from 158 to 152 minutes.

Figure 5.3(b) presents the minimum objective for a flight for which the user preference is to minimize discomfort. The minimum objective is less than $1.61*10^4$ from the first generation. This means a route free of discomfort is found from the first generation. For the next generations, GA searches for the solution with minimum time travelled. Since the discomfort is 0 and the user preference is 0.000001, the objective is rewritten as 0.000001*time. The minimum objective in 5.3(b) reduces from 0.000161 to 0.000152. This means time reduces by about 9 minutes from 161 minutes to 152 minutes. As the minimum discomfort is found from the first generation, GA will focus on searching for the minimum time solution which also has minimum discomfort. This explains why the generations where the best solutions are found for flights preferring to minimize discomfort are smaller than the ones where the best solutions are found for flights preferring to minimize time.

b) **Time and Discomfort Objectives of UPR flights given routes**

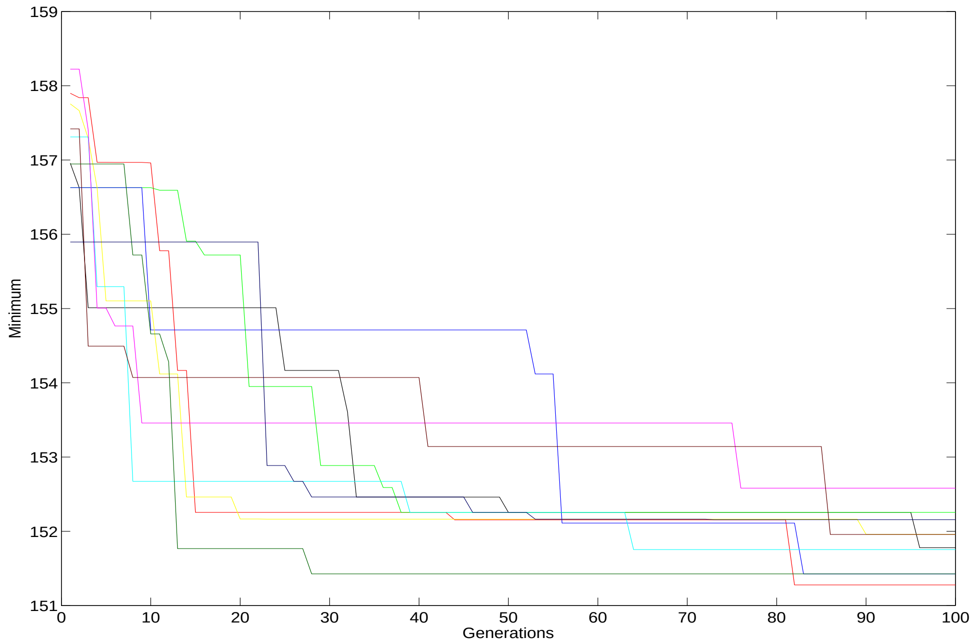
Figure 5.4 shows the results of time and discomfort objectives of runs for 7 flight scenarios, 4 weather scenarios and 10 different seeds. Flights are given routes.

Figure 5.4(b) shows that the algorithm can always find solutions to avoid almost all the bad weather areas. As shown in the figure, the sum of discomfort from all flights belonging to utility allocation UA0 (all preferring to minimize discomfort) is very near to 0. This happens even in the weather scenario affecting all the flights.

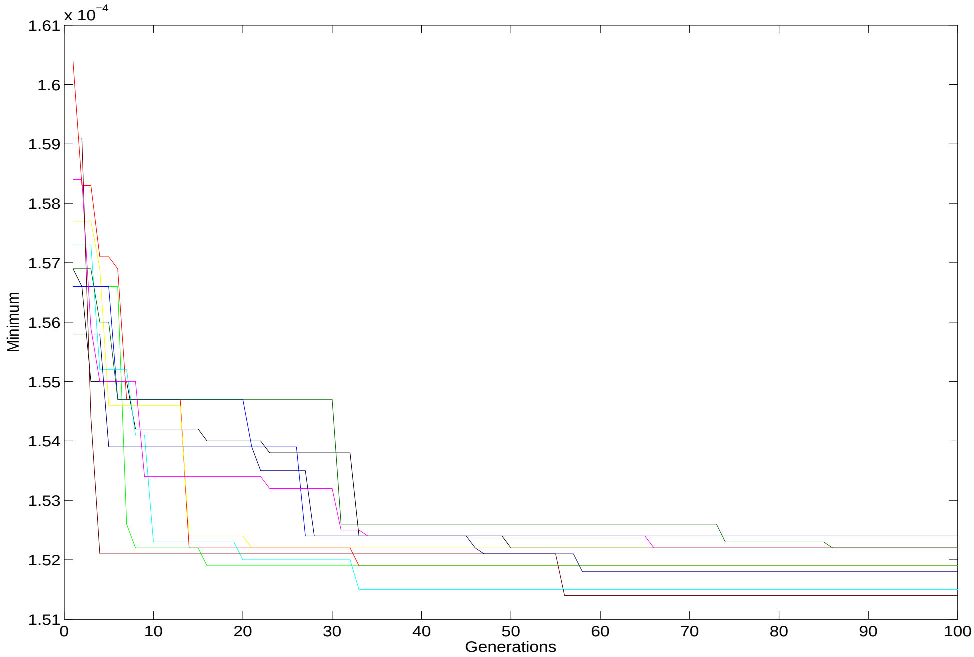
Figure 5.4 demonstrates that the times and discomforts of the best solutions provided by GA from different seeds for each flight-weather scenario behave as expected, and are very near to each other. The flight scenario UA0, in which all flights prefer to minimize discomfort, leads to the smallest discomfort and largest time. The flight scenario UA6, in which all flights prefer to minimize time, leads to the smallest time and largest discomfort.

The utility allocations UA2 and UA3 are neutral to the time and discomfort optimisations. UA2 has one third of flights for each of the 3 user preferences, and UA3 has all flights choosing the neural preference to optimise both time and discomfort. These utility allocations don't give a UPR method a strong message on which objective they prefer to minimize, so UPR methods vary in the UPR routes they choose for flights. This means the orders of UA2 and UA3 can change from one method to the other as well as from one problem to the other.

If UA2 and UA3 are discounted, the best order with respect to the time objective



(a) Minimum objective through generations of an example flight; The flight is given a route and prefers to minimize time; Each line presents the results for one seed.



(b) Minimum objective through generations of an example flight; The flight is given a route and prefers to minimize discomfort; Each line presents the results for one seed.

Figure 5.3: Minimum objective through generations of an example flight. The flight is given a route and prefers to minimize time or discomfort.

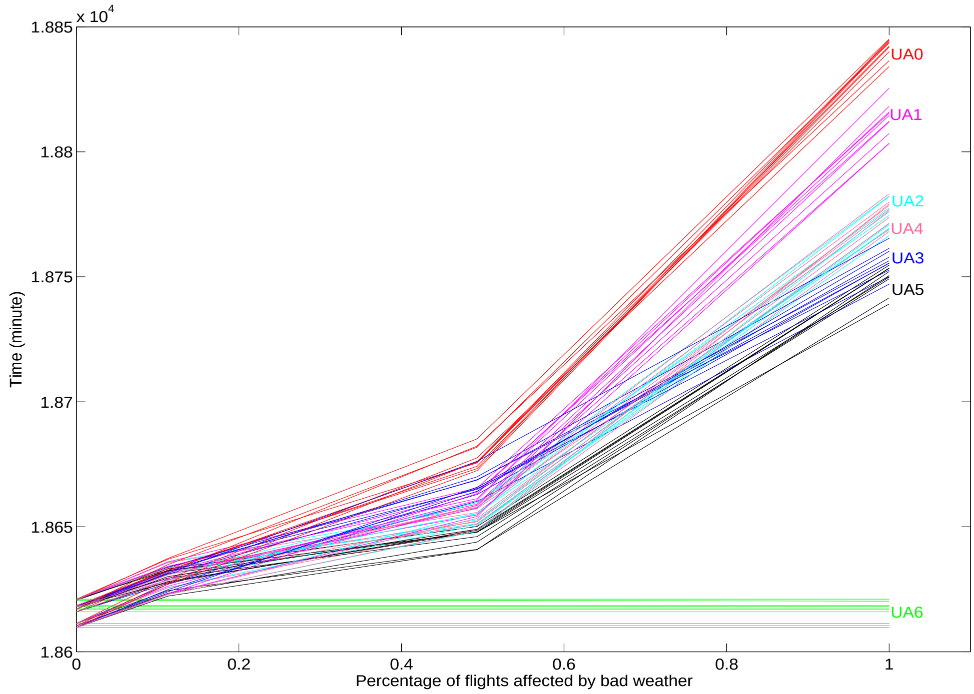
Utility Allocation	UP0 - 0.000001	UP1 - 0.5	UP2 - 0.999999	Pref Discomfort
UA0	1	0	0	1
UA3	0	1	0	1
UA1	0.5	0.3	0.2	0.8
UA4	0.2	0.5	0.3	0.7
UA2	0.33	0.33	0.33	0.66
UA5	0.2	0.3	0.5	0.5
UA6	0	0	1	0

Table 5.3: Flight utility allocation in the order of percentage of the flights preferring to minimize discomfort.

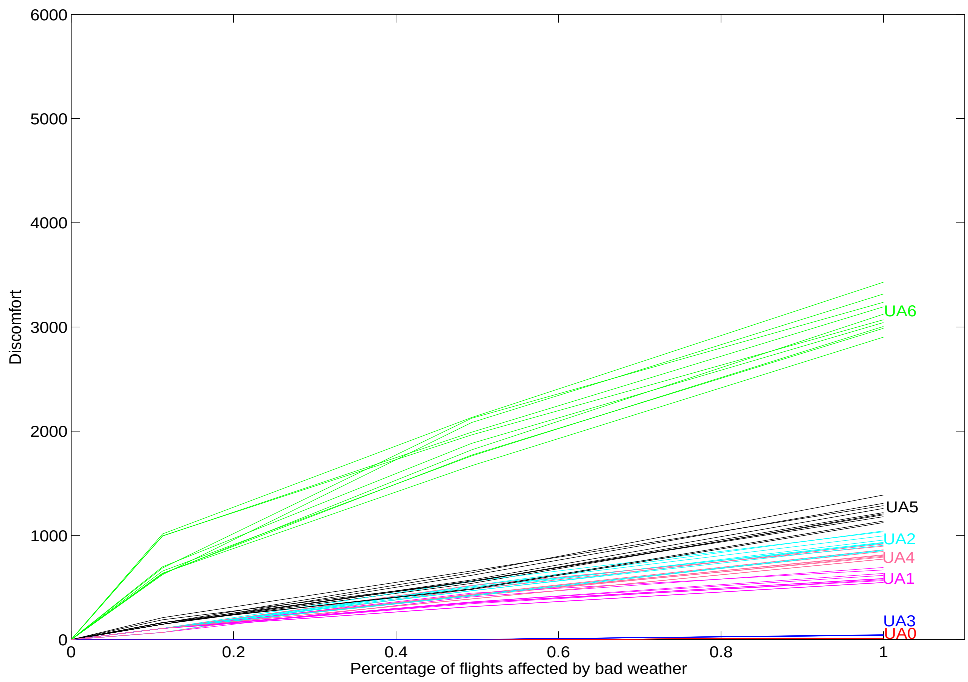
is UA6, UA5, UA4, UA1, UA0 and that of the discomfort objective is the reverse. We temporarily call these orders of the 5 utility allocations as "*the orders of 5*". These orders happen for all the experiments with the different problems and methods, as presented in the next sections. If UA2 and UA3 are counted, although they are not consistent, they still follow some reasonable patterns.

In Figure 5.4(b), when UA2 and UA3 are added, the better order of discomfort objective for the 7 utility allocations is UA0, UA3, UA1, UA4, UA2, UA5, and UA6. This corresponds to the pattern in Table 5.3. This table sums the two columns UP0 and UP1 of Table 3.3 and sorts the utility allocations first by this sum and then by UP0. The reason for ordering first by the sums of UP0 and UP1 is that both UP0 and UP1 prefer to minimize discomfort (UP0 extremely, and UP1 neutrally prefers). The reason for ordering next by UP0 is that if the sums of the two utility allocations are the same, the one with more flights that strongly prefer to minimize discomfort, will provide better discomfort.

Applying this sorting to the two column UP1 and UP2 for time preference, we have the order of utility allocations in Table 5.4 almost the same to that in Figure 5.4(a). There is only a swap order between UA3 and UA5. However UA3 and UA5 in the figure almost overlap each other.



(a) Time in different flight and weather scenarios; GA method and GRP problem.



(b) Discomfort in different flight and weather scenarios; GA method and GRP problem.

Figure 5.4: Time and discomfort in different flight and weather scenarios; GA method and GRP problem.

Utility Allocation	UP0 - 0.000001	UP1 - 0.5	UP2 - 0.999999	Pref Time
UA6	0	0	1	1
UA3	0	1	0	1
UA5	0.2	0.3	0.5	0.8
UA4	0.2	0.5	0.3	0.8
UA2	0.33	0.33	0.33	0.66
UA1	0.5	0.3	0.2	0.5
UA0	1	0	0	0

Table 5.4: Flight utility allocation in the order of percentage of the flights preferring to minimize time.

5.3.2 3-D User Preferred Routes using Genetic Algorithms for flights given cruising altitudes

In this section, a Genetic Algorithm is used to find the 3-D user preferred route for the GAP problem, where a flight is given a cruising altitude and the flight does not change its altitude during cruise phase.

The experiment design is presented in Section 5.3.2.1. The content and purpose of the experiment are the same as those for the GRP problem in Section 5.3.1.

Some results are presented in Section 5.3.2.2.

5.3.2.1 Experiment design

a) **Representation**

As UPR flights are given cruising altitudes, only the horizontal path of a flight needs to be optimised. Therefore, a chromosome here is represented by a series of horizontal segments for a flight. A horizontal segment is the horizontal projection of a segment. This segment may be a cruise segment in a cruise phase between Top Of Climb (TOC) and Top of Descent (TOD), or a climb segment in a climb phase from the origin to TOC, or a descent segment in an approach phase from TOD to the destination.

A horizontal segment is defined by the direction *dir* of an aircraft and the relative length l the aircraft will fly following the direction. The direction is the deviated

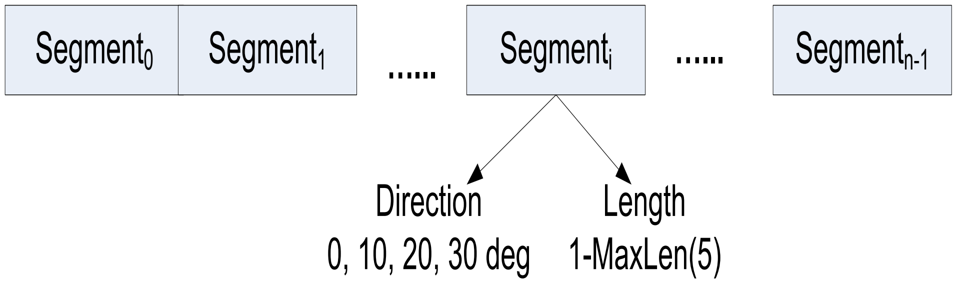


Figure 5.5: Chromosome for GA, flights are given routes

angle of the segment from the direct route from the current position to the destination. It can be 0, ±10, ±20, or ±30 degrees. The relative length l of a horizontal segment is an integer value from 1 to a predefined parameter $MaxLen$ (the maximum length of a horizontal segment). The absolute length (the distance) of a segment is $l * 100$ (km). Figure 5.5 shows the chromosome structure. This representation can ensure that the aircraft always flies towards its destination, as the result of the deviated angle constraint.

The parameters we used are: the maximum length of a horizontal segment $MaxLen$ is 5, the number of segments in a chromosome is $\frac{GCD}{100}$, where GCD is the great circle distance between the origin and destination of a flight measured in km.

b) **Genetic Operators**

We use the standard genetic operators as presented in Section 5.2. The mutation operator works similarly to that for GRP problem. The horizontal segment is mutated to become a new one by changing its direction, which is one of the values (0, ±10, ±20, or ±30 degrees) or its length, which is from 1 to the maximum length $MaxLen$.

c) **Generation of the 4-D trajectory**

Algorithm 7 generates the 4-D trajectory for a flight, which is given a cruising altitude, using a chromosome. First the route of 2-D points (the list of points) from the series of horizontal segments in the chromosome is achieved. The point list is initialized as empty. The origin airport is first added to the point list, and it is considered as the initial current point. At each step, a horizontal segment is chosen in sequence from the first one in the series of segments. The next point is determined, based on the distance and the direction from the current point to the next point (which are indicated in the segment). The next point is added to the point list and becomes the new current point. When the distance from the current point to the destination is less than a predefined distance, or all the segments have been gone through, the destination is added to the point list, and this process finishes.

The route, which is formed by the point list and indicates horizontal movement of the aircraft, is simulated to obtain a 4-D trajectory, using the given cruising altitude. The TOC (TOD) is determined first by climbing (descending) aircraft from origin (destination) to (from) the cruising altitude following the route (which is defined by the series of horizontal segments). Then the aircraft cruises from TOC to TOD following the route.

Algorithm 7 Generation of the 4-D trajectory for a flight given a cruising altitude, using a chromosome

```
{Get a 2-D (latitude, longitude) route, which is a series of waypoints, using the chromosome}
i=0
exit = false
The point list is empty
The current point is the origin airport
Add the current point into the point list
while i < number of segments and not exit do
    Calculate the distance from the current point to the next point(D). It is calculated as l * 100 (km), where l
    is the relative length of the horizontal segment_i.
    if The distance from the current point to the destination is less than D then
        Determine the ending point of the horizontal segment, based on the current point, the direction and
        distance of the horizontal segment.
        Add the ending point to the point list.
        The current point is the ending point.
    else
        Add the destination to the point list
        exit = true
    end if
end while
if exit is false then
    Add the destination to the point list
end if
{Simulate the trajectory, using the 2-D route and the given cruising altitude}
Algorithm 1 in Section 3.7.1 is used to simulate a 2-D route to climb to the target altitude, where the starting
point is the origin airport, and the target altitude is the given cruising altitude. The ending point of the
simulated segment is TOC.
Algorithm 3 in Section 3.7.7 is used to backward simulate the aircraft following a 2-D route to descend from
a starting point given altitude to an ending point, where the ending point is the destination and the starting
point is TOD whose altitude is the given cruising altitude.
Algorithm presented in Section 3.7.10 with algorithm flow in Figure 3.17 (which is used to simulate the aircraft
following a 2-D route to cruise from a starting point to an ending point) is used, where the starting point is
TOC and the ending point is TOD.
```

Figure 5.6 shows an example of the process to achieve a point list (a 2-D route) from a chromosome presenting horizontal movement for a flight whose the distance from the origin to the destination is 700 km. The number of elements in the chromosome is 7, which are presented in the top of the figure. An element includes two fields: the segment direction (the deviated angle of the segment from the direct route to the destination) and segment length. A direction can be ± 0, ± 10, ± 20, and ± 30 (deg). A segment length can be from 1 to 5. The

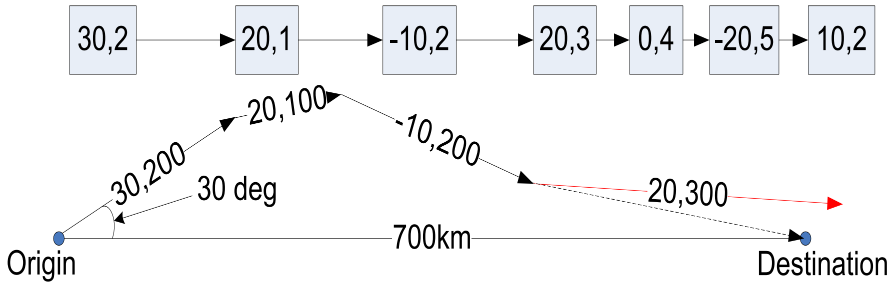


Figure 5.6: An example of simulation for chromosome representing a flight given a cruising altitude

first 3 segments tell aircraft to fly a distance of 200 km with a deviated angle of 30 (deg), then a distance of 100 km with a deviated angle of 20 (deg), and then a distance of 200 km with a deviated angle of -10 (deg). These distances are calculated as in (a). The distance of the forth segment is longer than the distance from the current point to the destination, so this segment and the last 3 segments are not used to navigate the aircraft, and the aircraft flies directly to the destination. The point list includes the origin, and the ending points of the segments (which are added in sequence).

The 3-D route is obtained by dropping the time dimension from the 4-D trajectory.

d) **Evaluation**

The objective function used to evaluate a flight is given by by Equation 5.1.

e) **Mutation probability investigation and parameter settings**

Mutation probability investigation

The experiment to investigate mutation probability is the same as for the GRP problem. The percentage of flights for which each mutation probability is the best is presented in Table 5.5. The table again shows that the mutation probability of 0.1 is the best choice in general. The percentage of flights for which the mutation of 0.1 is the best is 44.4% while those for the mutation probabilities of 0 (local search) and 0.6 (random search) are only 10.6% and 18.3% respectively.

Mutation probability	0	0.1	0.2	0.3	0.4	0.5	0.6
Percentage of flights	10.6%	44.4%	35.2%	30.3%	25.4%	16.9%	18.3%

Table 5.5: Mutation probability investigation; GA for flights given cruising altitudes

User Pref.	BW0 - 0%	BW1 - 10%	BW2 - 50%	BW3 - 100%
Discomfort	NaN	29	34	41
Neutral	NaN	54	59	62
Time	53	53	53	54

Table 5.6: The average of generations which has the best individual, GA for flights given cruising altitudes.

5.3.2.2 Results

a) **Generation of best individuals and the minimum objective through generations**

Table 5.6 shows the average number of generations at which the best individuals are found. In the table, the row and column headings have the same meanings as in the previous section. The results show that GA can reach its best solution more quickly for flights preferring to minimize discomfort than for flights preferring to minimize time (in about 35 and 55 generations respectively).

Figure 5.7 shows the minimum objective through generations of an example flight. The flight has the chromosome size of 20 (the average size of chromosomes). It is given a cruising altitude and prefers to minimize time or discomfort.

Figure 5.7(a) presents the minimum objective for a flight for which the user preference is to minimize time. As the discomfort is multiplied by a very small number 0.000001, the minimum objective can be considered to present time objective. The figure shows that the objective values from 10 different seeds in the final generation are around 172 minutes (only about 1 minute different from each other). They reduce from 177 to 172 minutes overall.

Figure 5.7(b) presents the minimum objective for the flight when the user preference is to minimize discomfort. The minimum objective is less than $1.79*10^4$ from the first generation. This means a route free of discomfort is found from the first generation. For the next generations, GA searches for the solution with minimum time travelled. Since the discomfort is 0 and the user preference is 0.000001, the objective is rewritten as 0.000001*time. The minimum objective in 5.7(b) reduces from 0.000179 to 0.000172. This means time reduces by about 7 minutes from 179 minutes to 172 minutes. As the minimum discomfort is found from the first generation, GA will focus on searching for the minimum time solution which also has minimum discomfort. This explains why the generations where the best solutions are found for flights preferring to minimize discomfort are smaller than the ones where the best solutions are found for flights preferring to minimize time.

b) **Time and Discomfort Objectives of UPR flights given cruising altitudes**

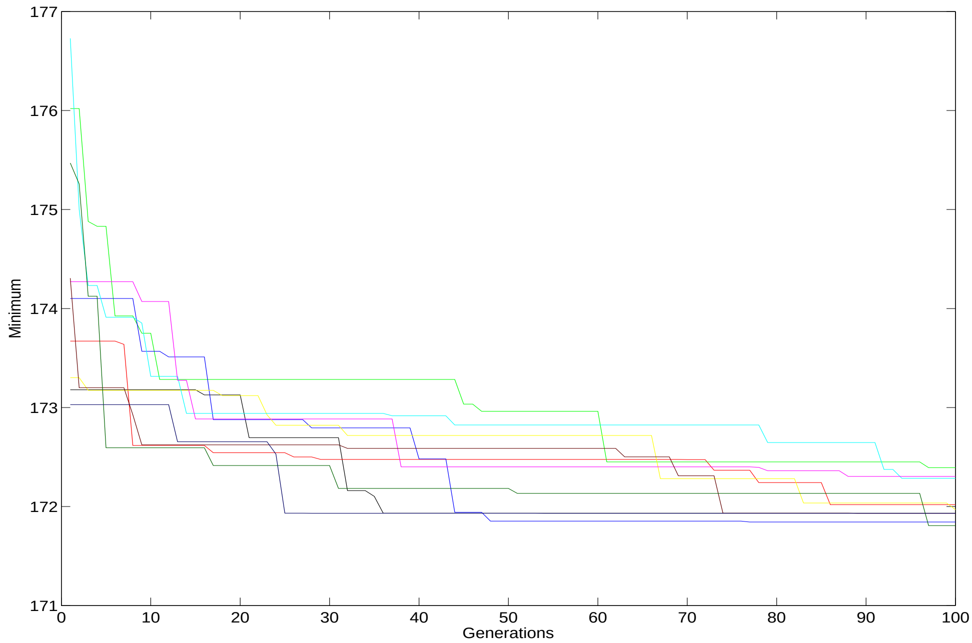
Figure 5.8 shows the results of time and discomfort objectives of runs for 7 flight scenarios, 4 weather scenarios and 10 differs seeds. Flights are given cruising altitudes.

The better orders of time and discomfort objectives follow "*the orders of 5*". The discomfort order of the 7 utility allocations in Figure 5.7 is the same as that for the GRP problem in Table 5.3. The better order of time objective follows the same pattern, which sorts the utility allocations by column UP2 (which extremely prefers to minimize time).

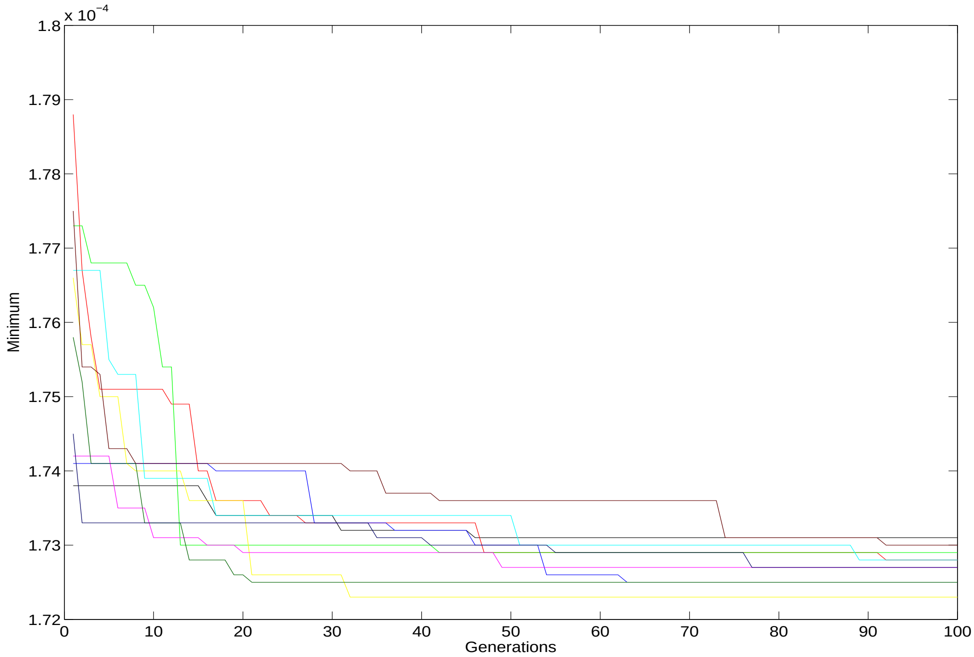
5.3.3 Conclusion

The designed GA is able to find UPR routes for UPR flights which are given either routes or cruising altitudes. The mutation probability investigation shows that GA can work better than local search (corresponding to the mutation rate of 0), and random search (corresponding to the mutation rate of 0.6). Furthermore the solutions provided by the designed GA are similar to those provided by the algorithm using control point networks, which will be presented in Chapter 6.

The algorithm can provide URP routes satisfying user preference. For example it can provide discomfort-free routes for flights preferring to minimize discomfort,

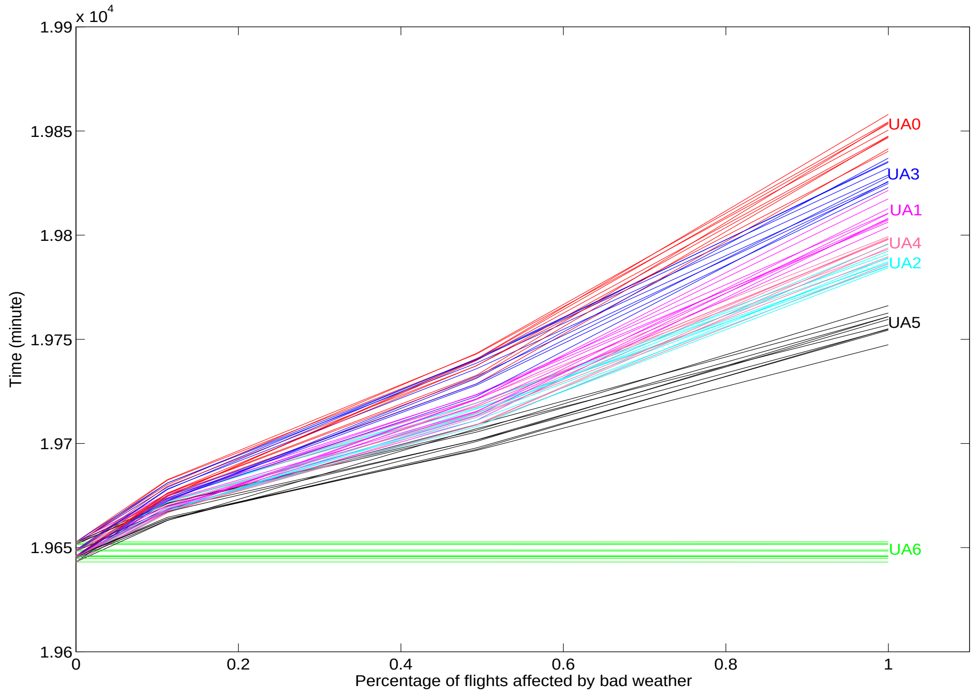


(a) Minimum objective through generations of an example flight; The flight is given a cruising altitude and prefers to minimize time; Each line presents the results for one seed.

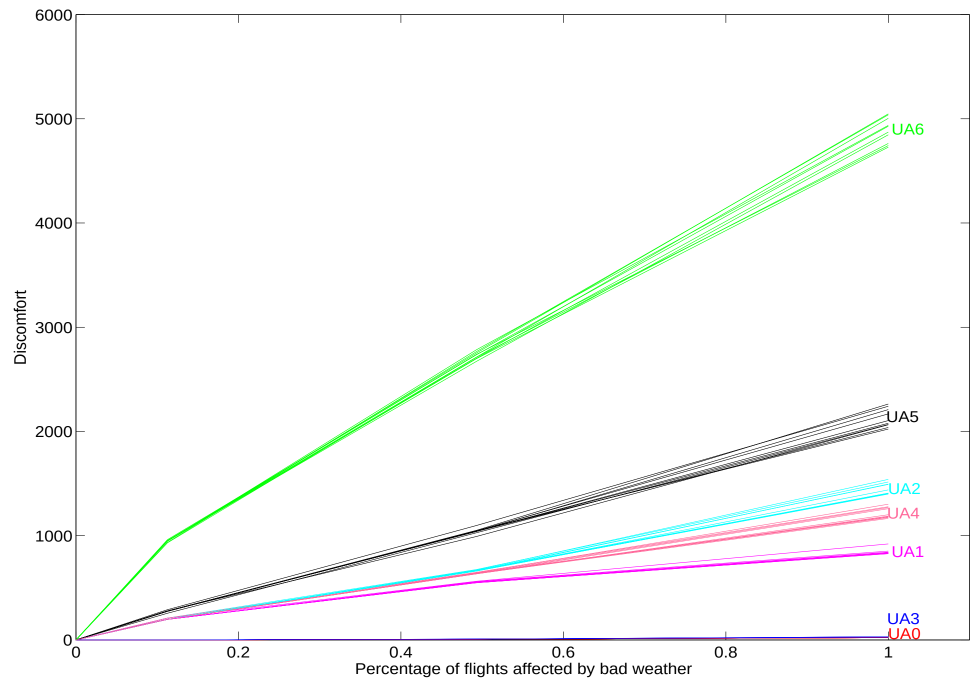


(b) Minimum objective through generations of an example flight which is given a cruising altitude and prefers to minimize discomfort; Each line presents the results for one seed.

Figure 5.7: Minimum objective through generations of an example flight. The flight is given a cruising altitude and prefers to minimize time or discomfort.



(a) Time in different flight and weather scenarios; GA method and GAP problem.



(b) Discomfort in different flight and weather scenarios; GA method and GAP problem.

Figure 5.8: Time and discomfort in different flight and weather scenarios; GA method and GAP problem.

even in the worst weather scenario which affects all the flights. This is understandable: as the bad weather is not very dense in reality, a discomfort free route can always be found. Using the utility (weighted sum) can help to minimize not only one objective but both objectives: discomfort, and time. The results show that the best solutions for flights preferring to minimize discomfort can be found faster than the ones for flights preferring to minimize time. The reason is the minimum discomfort route is usually found quickly. Then GA can focus on improving the time objective of solutions whose discomforts are minimum.

5.4 3-D User Preferred Routes using Learning Classifier System

In this section, an efficient design of a Learning Classifier System for multi-flight navigation is presented for each problem (GRP and GAP). A classifier is represented by a set of ordered rules, which are used to simultaneously navigate all the flights in the airspace. Navigation of a flight is based on the relation of the flight to factors of the air traffic environment, such as wind and bad weather. This system continually learns and refines the rules of classifiers by a Genetic Algorithm to discover the classifier which navigates flights with minimal time of flying and minimal discomfort.

In this section, we do experiments to show that LCS can find good solutions according to the objectives of time, and discomfort. Furthermore, by showing the classifier in symbolic representation we demonstrate the decision rules used in the classifier. The rules are transparent to the user, making this a “white box” approach to finding UPR routes.

5.4.1 Algorithm

The algorithm evolves a population of classifiers, using a Genetic Algorithm similar to Algorithm 4. The difference is that an individual in these algorithms is a classifier. The population of classifiers is evolved over time in order to find the best classifiers for flight navigation. Each classifier is a set of normal rules and one default rule. A normal rule includes two parts: condition and action. The condition of a normal rule is defined by a relative wind direction with a flight direction and a bad weather level. The relative wind direction may be either tail, cross up, cross down, or head (it is determined as in Figure 5.11). The bad weather level may be none, low, medium-low, medium, medium-high, or high. The action of a rule defines the segment type (climb, cruise, descent) or the turning angle (0, ± 10, ± 20, ± 30 (deg)) and the additional altitude or additional cruising distance an aircraft needs to execute.

5.4.2 3-D User Preferred Routes using Learning Classifier System for flights given 2-D routes

In this section, a Learning Classifier System is designed to find 3-D user preferred routes for the GRP problem, where flights are given 2-D (latitude, longitude) routes and the flights can change their altitudes during cruise phase.

Section 5.4.2.1 presents the experiment design. The classifier representation is presented first, followed by genetic operators. Then the algorithm to generate 4-D trajectories for flights using a classifier is presented, followed by the evaluation of a classifier. The purpose of the experiment is to show that LCS can find optimal routes for UPR flights and it can provide solutions understandable to users.

The results are presented and discussed in Section 5.4.2.2.

5.4.2.1 Experiment design

a) **Classifier Representation**

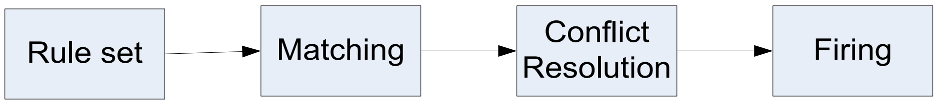


Figure 5.9: The process of choosing a rule

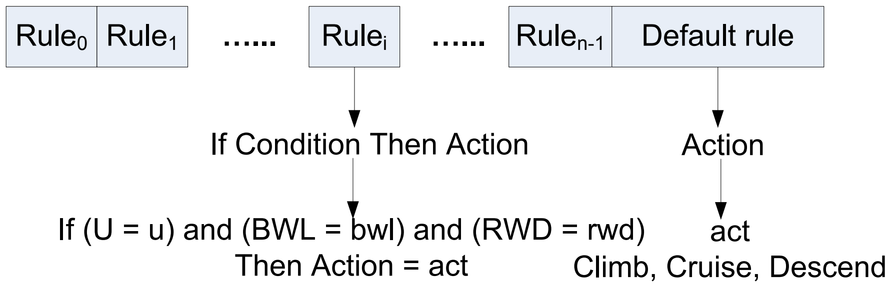


Figure 5.10: Classifier for flights are given routes

In a learning classifier system, the process of choosing a rule in a set of rules is described in Figure 5.9. The "Matching" block finds a sub-set of the rules in the set that satisfy a given condition. The rules in the sub-set can conflict with each other. This requires some techniques to choose which rule for use. This problem is solved by the "Conflict Resolution" block. The action of the chosen rule is then executed in the "Firing" block. In this thesis, we design a classifier as a set of ordered rules. When there are a number of matched rules in a classifier, the first matching rule in the order is used.

A classifier with a set of normal rules and one default rule is presented in Figure 5.10. One normal rule has two parts: condition and action. The default rule does not have a condition part, it has an action part only.

Table 5.7 lists the parameters (condition and action) of a normal rule and their values, where U is the utility, BWL is the bad weather level, RWD is the relative wind direction, and ACT is the action. This rule is interpreted as "if (U = u) and (BWL = bwl) and (RWD = rwd) then ACT = act".

- u can be 0.000001, 0.5, or 0.999999, representing time, neutral, and discomfort preferences respectively (as presented in Section 3.9.2).
- bwl is an integer from 0 to 5, representing $none$, low, $medium - low$, $medium$, $medium - high$, or $high$ respectively.

Parameter	U	BWL	RWD	ACT
Value	u	bwl	rwd	act

Table 5.7: Rule parameters and values

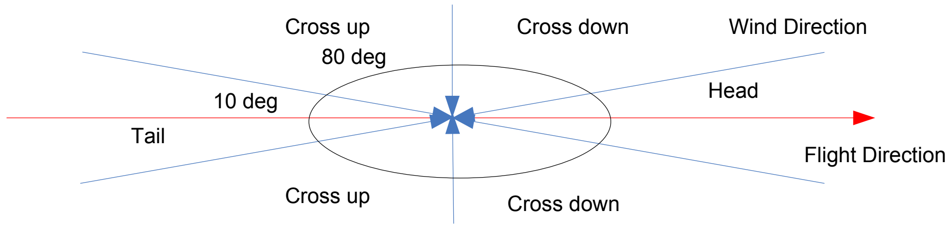


Figure 5.11: Relative Wind Direction with Flight Direction

- rwd is an integer from 0 to 4, representing $none$, $tail$, $crossup$, $crossdown$, or $head$ respectively (see Figure 5.11).
- act tells aircraft to climb, or descend a predefined additional altitude, or cruise a predefined distance.

The parameters we used are: the predefined additional altitude of a climb/descent segment is 1000 feet, the predefined distance of a cruise segment is 100 km, the number of rules in one classifier is 51, including 50 normal rules and one default rule.

b) **Genetic Operators**

Tournament selection is used to select classifiers. The crossover operator crosses two selected classifiers $parent1$ and $parent2$ to create two new classifiers $child1$ and $child2$. Firstly, it generates a random number $rand$. If $rand$ is less than the predefined crossover probability $pcross$, the algorithm will use the cross over point $site$ generated randomly from 0 to $nrule - 1$ to cross $parent1$ and $parent2$. If $rand$ is higher than $pcross$, $child1$ and $child2$ will be the clone of $parent1$ and $parent2$ respectively.

In general, the mutation operator works similarly to that of the Genetic Algorithm for the GRP problem. In order to mutate a rule, an integer number mp from 0 to 3 is generated randomly, which indicates the mutated part of the rule. If mp is 0, the utility of the rule is mutated. If mp is 1, the bad weather level of the rule is mutated. If *mp* is 2, the relative wind direction of the rule is mutated. If *mp* is 3, the action of the rule is mutated. The mutation operator also applies to default rules.

c) **Generating 4-D trajectories**

A classifier is evaluated by first obtaining all the 4-D trajectories of flights using the classifier through simulation.

A flight starts from its origin. In order to determine the segment for the flight to move, firstly the rule to navigate the flight is chosen. If the flight at the current state satisfies a normal rule in the classifier, this rule is chosen to navigate the flight, otherwise the action of the default rule is used. The action can tell the flight to climb, cruise, or descend. The first segment must be a climb segment. The execution of an action makes a climb, cruise, or descent segment. The additional altitude of a climb and descent segment, and the distance of a cruise segment are predefined parameters as presented above. The algorithm to simulate the series of segments to obtain the 4-D trajectory is similar to that with two parts in Algorithms 5 and 6.

The 3-D routes are obtained by removing the time dimension from the 4-D simulated trajectories.

d) **Evaluation**

The objective of the simulated route is calculated by equation 5.1. The classifier's fitness is the sum of all the objectives of flights.

5.4.2.2 Results

a) **Generation of best individuals and the minimum objective through generations**

Table 5.8 shows the average number of generations at which the best individuals are found. The cells at column "BW0 - 0%" and rows from "UA0" to "UA6" are

Utility Allocation	BW0 - 0%	BW1 - 10%	BW2 - 50%	BW3 - 100%
UA0	NaN	37	55	56
UA1	NaN	69	68	70
UA2	NaN	72	67	70
UA3	NaN	11	40	64
UA4	NaN	74	72	65
UA5	NaN	76	82	63
UA6	1	20	63	58

Table 5.8: The average of generations where the best individual is found; LCS for flights given routes.

"NaN", as we chose only one flight scenario "UA6" on which all flights prefer to minimize time to experiment with weather scenario "BW0 - 0%".

The results show that LCS can reach its best solution quickly for the simplest flight scenarios: "UA0", and "UA6" are faster than other flight scenarios. In these flight scenarios, all flights have the same user preference (to minimize time or discomfort). In the other flight scenarios, both user preferences are considered. In the no-bad weather scenarios, the best classifier is found in the first generation as shown at the cell of column "BW0 - 0%" and row "UA6". In this classifier, the action of rules, which are used to navigate flights, are climb or cruise. This means flights tend to reach high altitude by climbing. Then they cruise to TOD and continuously descend to the destination.

Figures 5.12(a) and 5.12(b) show the minimum objective through the generations for flight scenarios UA6, UA0.

Figure 5.12(a) (UA6) shows that LCS can reduce time travelled by about 14 minutes from 19382 (in the first generation) to 19368 minutes (in the last generation). Although LCS can minimize time for user preferred routes, the best classifier is not significantly improved from the first generation to the last generation. This is explained as follows. Given a 2-D fixed route, what a flight can do to optimise time is to choose flight levels that has favorable wind and at which it can perform best with its aircraft performance. However winds at flight levels where aircraft can perform well (normally at higher altitudes), are not much different. Therefore the relative wind direction condition of a rule does not affect much on the time objective. That is why the time objective is not improved significantly, while classifiers in general and the relative wind direction conditions are evolved.

However LCS for discomfort minimization can be improved noticeably. Figure 5.12(b) (UA0) shows that LCS can help to reduce discomfort from 1320 to 1140. This means LCS can reduce the time period flights pass through bad weather areas by 60 minutes from about 440 to 380 minutes if all the bad weather levels are 3 (the average bad weather level).

b) **Time and Discomfort Objectives of UPR flights given routes**

Figure 5.13 shows the results of time and discomfort objectives of runs for the combinations of 7 flight scenarios, 4 weather scenarios, and 10 different seeds. Flights are given routes.

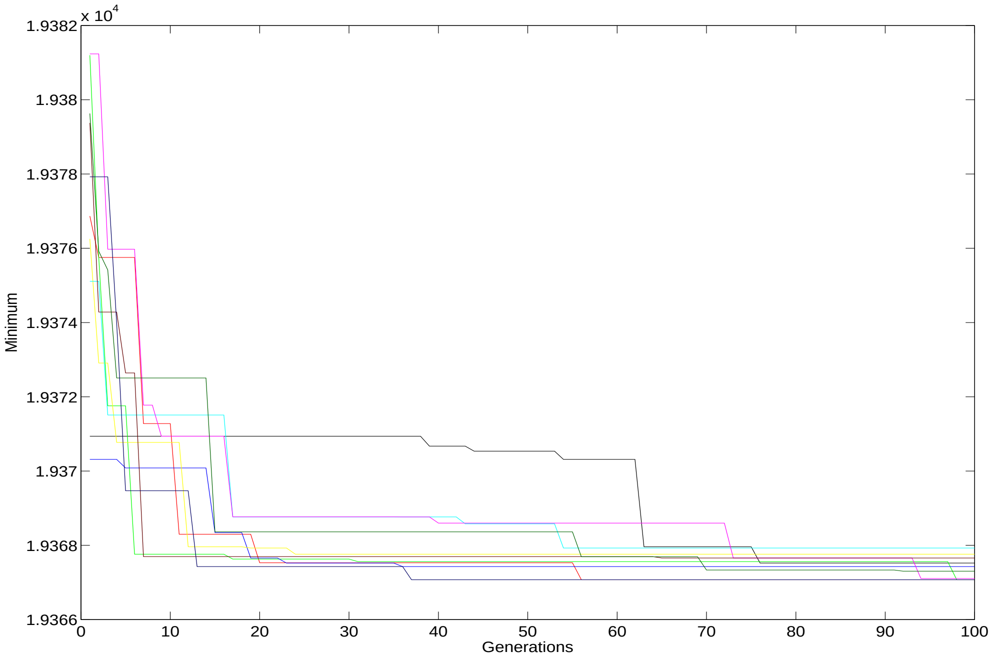
Figure 5.13 demonstrates that the times and discomforts of the best solutions provided by LCS from different seeds for a flight-weather scenario are very near to each other. LCS can provide the same optimal time in all 10 runs for flight scenario UA6 (all flights prefer to minimize time). The flight scenario UA0, where all flights prefer to minimize discomfort has smallest discomfort and largest time, while the flight scenario UA6, where all flights prefer to minimize time has smallest time and largest discomfort.

However, Figure 5.13(b) shows that the LCS cannot help flights to avoid all the bad weather areas as algorithm using GA can do. This happens in all weather scenarios with bad weather cells.

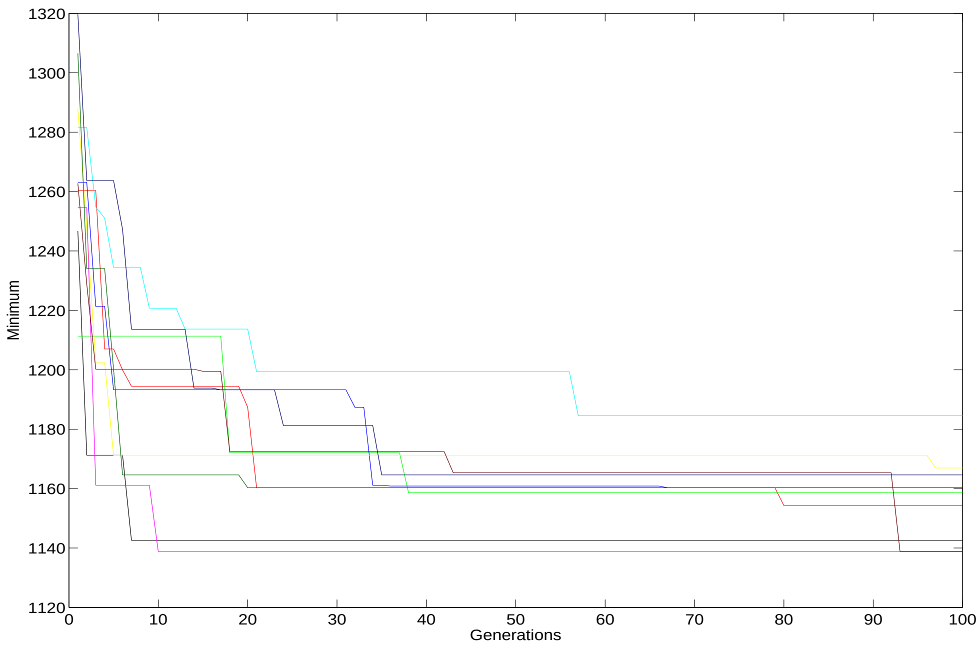
The better orders of time and discomfort objectives in Figures 5.13(a) and 5.13(b) follow "*the orders of 5*". The discomfort order for the 7 utility allocations in Figure 5.7 is the same as in Table 5.3. Figure 5.13(a) shows that LCS can optimise well for the neutral utility allocations UA2, and UA3.

c) **An example classifier**

Table 5.9 presents the best classifier for the bad weather scenario affecting 50%

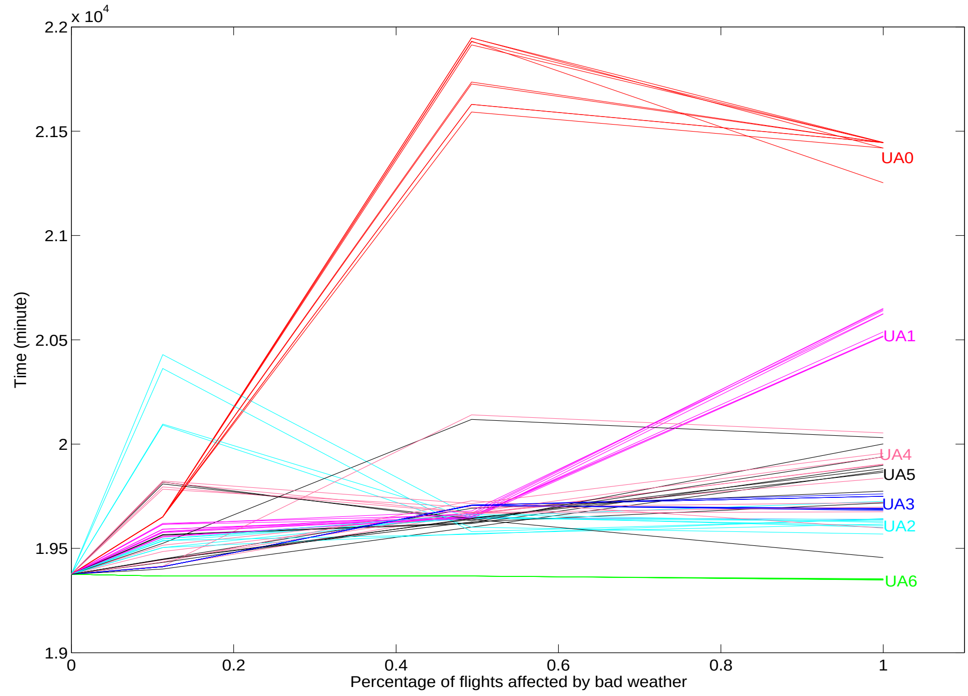


(a) Minimum objective through generations; Flights are given routes and prefer to minimize time; Each line presents the results for one seed.

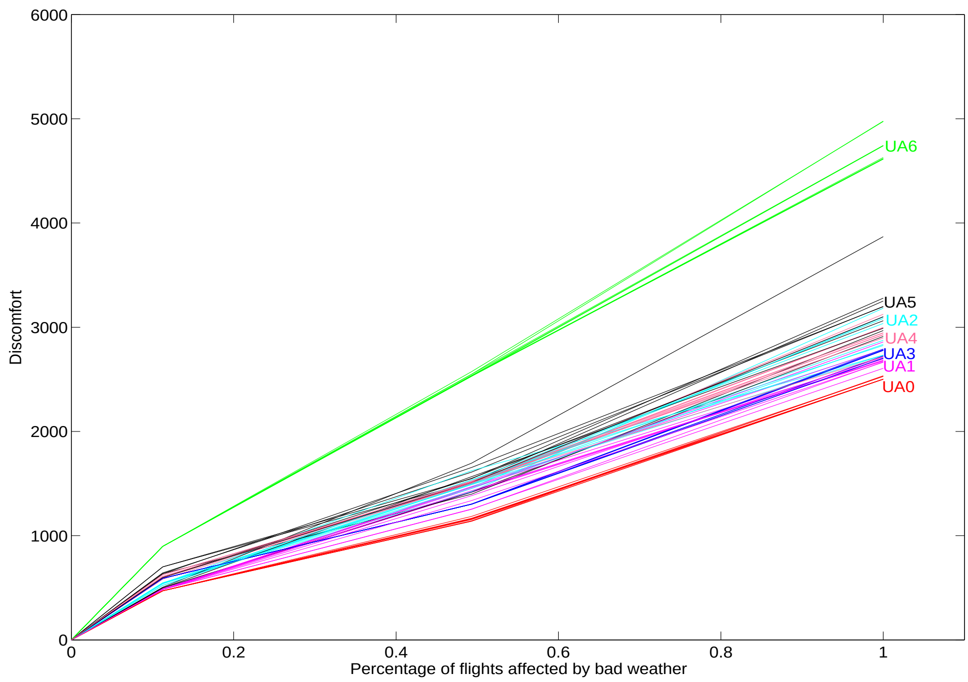


(b) Minimum objective through generations; Flights are given routes and prefer to minimize discomfort; Each line presents the results for one seed.

Figure 5.12: Minimum objective through generations; Flights are given routes and prefer to minimize discomfort or time.



(a) Time in different flight and weather scenarios; LCS method and GRP problem.



(b) Discomfort in different flight and weather scenarios; LCS method and GRP problem.

Figure 5.13: Time and discomfort in different flight and weather scenarios; LCS method and GRP problem.

of flights and the flight scenario UA2. In this flight scenario, the 3 types of preferences (discomfort, neutral, and time) have the same number of flights (33% of flights). In the table "Rule ID" is the index of a rule, "Usage" is the number of times a rule is used, and "Content" is the content of a rule. Since the classifier is a set of ordered rules, rules whose conditions are the same to a previous rule are removed from the classifier. That is why the classifier does not present all 50 normal rules.

Generally the action to climb is used the most: it is used in 717 times, while those for cruise and descend are used 238 and 166 times respectively in the classifier. This means the aircraft tend to reach high altitude to avoid bad weather as well as to achieve high ground speed. The number of climb actions is higher than the number of descent actions, as continuous descent is applied from TOD to destination.

From the classifier we can also see a number of rules using tactical maneuvers to avoid unfavorable weather or to take advantage of favorable weather condition. For example, rule 28: "IF U = time AND BWL = none AND RWD = cross up THEN cruise" and rule 6: "IF U = time AND BWL = none AND RWD = tail THEN cruise": these rules mean an aircraft will keep the current altitude (cruise) if the aircraft prefers to minimize time, there is no bad weather, and the favorable wind helps to increase the ground speed of the aircraft (cross up or tail wind). Rule 40 has the content of "IF U = comfort AND BWL = high AND RWD = cross up THEN descend". This rule means if an aircraft prefers to minimize discomfort and the bad weather level is high, the aircraft will change its altitude to avoid the bad weather. The other rules work as strategic planning. By using these rules aircraft can reach favorable weather and wind to achieve global objective.

Table 5.10 shows the frequency of using commands: climb, cruise, and descend for flights with comfort, neutral, and time preferences. It shows that flights with comfort preference execute a lot of climb and descend commands, as these commands help them to avoid the bad weather areas. Flights with neutral preference

Rule ID	Usage	Content
0	0	IF U = neutral AND BWL = med-high AND RWD = none THEN descend
1	0	IF U = neutral AND BWL = low AND RWD = cross down THEN cruise
2	0	IF U = time AND BWL = med-low AND RWD = none THEN climb
3	62	IF U = time AND BWL = none AND RWD = cross down THEN climb
4	2	IF U = time AND BWL = low AND RWD = cross up THEN descend
5	0	IF U = time AND BWL = med-high AND RWD = tail THEN descend
6	24	IF U = time AND BWL = none AND RWD = tail THEN cruise
7	0	IF U = time AND BWL = med-low AND RWD = cross down THEN descend
8	2	IF U = neutral AND BWL = med-low AND RWD = cross up THEN cruise
9	47	IF U = neutral AND BWL = none AND RWD = head THEN descend
10	0	IF U = neutral AND BWL = low AND RWD = head THEN descend
11	1	IF U = time AND BWL = med-high AND RWD = cross up THEN descend
12	42	IF U = comfort AND BWL = none AND RWD = none THEN climb
13	0	IF U = comfort AND BWL = medium AND RWD = cross down THEN descend
14	0	IF U = comfort AND BWL = medium AND RWD = head THEN cruise
15	0	IF U = time AND BWL = med-low AND RWD = head THEN climb
16	0	IF U = neutral AND BWL = med-low AND RWD = cross down THEN descend
17	0	IF U = time AND BWL = medium AND RWD = none THEN cruise
18	3	IF U = comfort AND BWL = med-high AND RWD = cross up THEN cruise
21	0	IF U = time AND BWL = high AND RWD = head THEN descend
22	147	IF U = neutral AND BWL = none AND RWD = cross down THEN climb
23	0	IF U = comfort AND BWL = low AND RWD = none THEN climb
25	0	IF U = time AND BWL = none AND RWD = none THEN cruise
26	1	IF U = neutral AND BWL = medium AND RWD = cross up THEN cruise
27	25	IF U = neutral AND BWL = none AND RWD = tail THEN cruise
28	183	IF U = time AND BWL = none AND RWD = cross up THEN cruise
29	0	IF U = comfort AND BWL = medium AND RWD = none THEN cruise
30	0	IF U = time AND BWL = high AND RWD = cross down THEN descend
31	55	IF U = comfort AND BWL = none AND RWD = tail THEN climb
32	0	IF U = time AND BWL = medium AND RWD = tail THEN climb
39	0	IF U = time AND BWL = low AND RWD = head THEN climb
40	108	IF U = comfort AND BWL = high AND RWD = cross up THEN descend
41	0	IF U = time AND BWL = low AND RWD = cross down THEN descend
44	0	IF U = comfort AND BWL = med-low AND RWD = head THEN descend
45	8	IF U = time AND BWL = med-low AND RWD = cross up THEN descend
48	0	IF U = neutral AND BWL = medium AND RWD = head THEN descend
default	411	climb

Table 5.9: An example of classifier for flights given routes

	Climb	Cruise	Descend
Comfort	46.63%	1.44%	51.92%
Neutral	66.22%	12.61%	21.17%
Time	22.14%	73.93%	3.93%

Table 5.10: Frequency of using commands for flights given cruising altitudes

do not climb and descend as much as flights preferring comfort. Flights with time preference do not make many climb and descend segments. Most of their commands are cruise commands. This means that these flights after climbing to optimal altitudes tend to keep cruising until approach phase.

5.4.2.3 Summary

The designed LCS for the GRP successfully discovered rules in all runs to optimise the system performance. LCS can help to optimise both discomfort and time. The better orders of discomfort and time objectives for different utility allocations are reasonable. The discomfort objective can be improved significantly through generations. The time objective is not improved much though generations due to the character of the problem, where the 2-D routes of flights are fixed. A number of discovered rules in the best classifier for tactical maneuvers agree with what happens in reality.

5.4.3 3-D User Preferred Routes using Learning Classifier System for flights given cruising altitudes

In this section a Learning Classifier System is designed to find 3-D user preferred routes for the GAP, where flights are given cruising altitudes and flights do not change their altitude during cruise phase.

The experiment and results are presented the same as those for the GRP problem in Section 5.4.2.

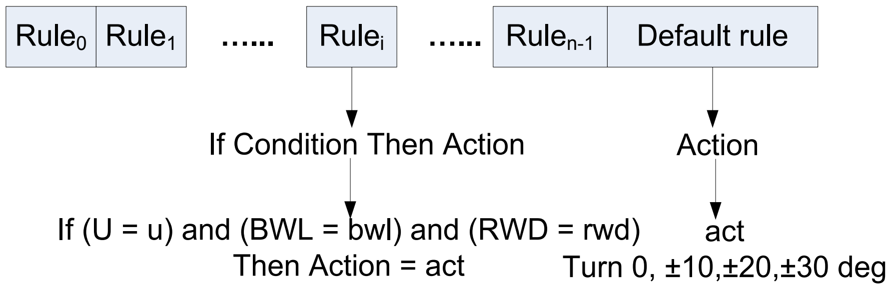


Figure 5.14: Classifier for flights given cruising altitudes

5.4.3.1 Experiment design

a) **Classifier Representation**

A classifier is a set of ordered rules including normal rules and one default rule, which are similar to those of a classifier designing for GRP. Figure 5.14 shows the classifier structure. The classifier structure is similar to that for the GRP problem in Section 5.4.2.1. The one difference is in the action of rule. The action tells an aircraft the direction to move horizontally. ACT is defined by a turn angle of -30, -20, -10, 0, +10, +20, or +30 (deg) (which is represented by an integer number 0, 1, 2, 3, 4, 5, or 6), and a predefined length of a segment. The turn angle of 0 (deg) means an aircraft flies straight from its current point to its destination. The turn angle less than 0 (deg) means an aircraft turns left. The turn angle greater than 0 (deg) means an aircraft turns right.

b) **Genetic Operators**

Genetic operators are the same as those for the GRP problem in Section 5.4.2.1.

c) **Generation of 4-D trajectories**

In order to obtain the 4-D trajectory of a flight, firstly a set of horizontal segments is determined, using the classifier. The flight starts from its origin and it is considered as the current point of the aircraft. If the distance from the current point of the aircraft to the destination is less than the given distance of a horizontal segment, the aircraft will fly directly to the destination. Otherwise, the rule to navigate the flight is chosen. If the flight at the current state satisfies

Utility Allocation	BW0 - 0%	BW1 - 10%	BW2 - 50%	BW3 - 100%
UA0	NaN	5	62	66
UA1	NaN	79	79	76
UA2	NaN	77	75	65
UA3	NaN	47	78	87
UA4	NaN	56	80	76
UA5	NaN	73	61	77
UA6	2	3	67	68

Table 5.11: The average of generations which has the best individual; LCS for flights given cruising altitudes.

a normal rule in the classifier, this rule is chosen to navigate the flight, otherwise the action of the default rule is used. At each step, a horizontal segment, which is a direct segment to the destination or a horizontal segment indicated in the chosen rule is used to navigate the flight. The algorithm to construct and simulate the series of horizontal segments to obtain the 4-D trajectory is similar to Algorithm 7.

d) **Evaluation**

As in Section 5.4.2.1, the objective of the simulated trajectory is calculated by Equation 5.1. A classifier's fitness is the sum of all the objectives of flights.

5.4.3.2 Results

a) **Generation of best individuals and the minimum objective through generations**

Table 5.11 shows the average number of generations at which the best individuals are found. As with the GRP problem, LCS can reach its best solution quickly for the simplest flight scenarios: "UA0", and "UA6" are faster than other flight scenarios. In the no-bad weather scenario "BW0 - 0%", the best classifier is found in the second generation as shown at the cell of column "BW0 - 0%" column and row "UA6". In this classifier, the action of all rules is to fly straight. This means aircraft fly great circle routes between their origins and destinations.

Figures 5.15(a) and 5.15(b) show the minimum objective through generations for flight scenarios UA6 and UA0 respectively.

Figure 5.12(a) (UA6) shows that LCS can reduce time travelled by about 120 minutes, from 19800 to 19680 minutes. Although the improvement (120 minutes) is still small, it is significantly bigger than for the GRP problem (only 14 minutes). This is explained as follows. Given a cruising altitude, when a flight chooses different routes to fly, the wind impact can be noticeably different as the results of changing wind directions. Therefore the relative wind direction condition of a rule affects the time objective.

Figure 5.15(b) (UA0) shows that LCS can help to reduce discomfort from 250 to 100. This means the algorithm can reduce the time period flights pass through bad weather areas by 50 minutes from about 83 to 33 minutes, if all the bad weather levels are 3 (the average bad weather level).

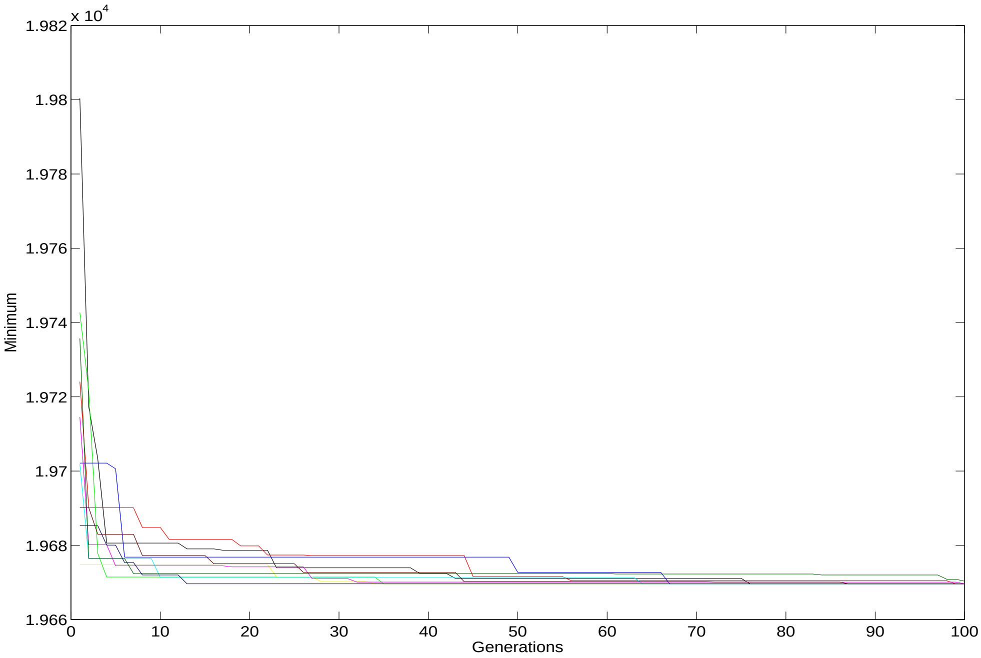
b) **Time and Discomfort Objectives of UPR flights given cruising altitudes**

Figure 5.16 shows the results of time and discomfort objectives of runs for the combinations of 7 flight scenarios, 4 weather scenarios, and 10 different seeds. Flights are given cruising altitudes.

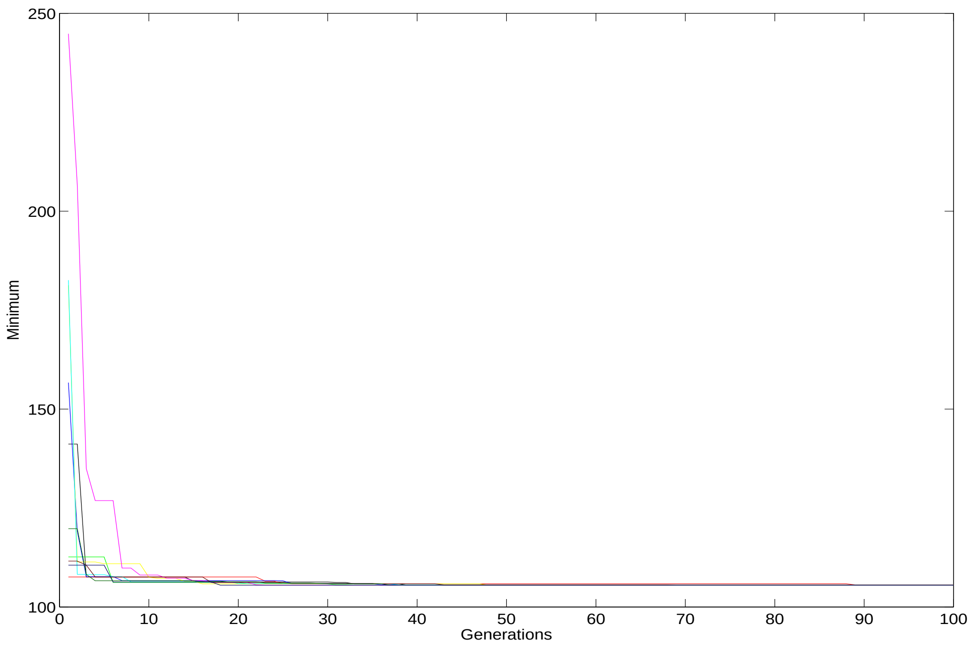
Figure 5.16 demonstrates that the times and discomforts of the best solutions provided by LCS from different seeds for each flight-weather scenario are very near to each other. LCS can provide the same optimal time for flight scenario UA6 (all flights prefer to minimize time). The flight scenario UA0, where all flights prefer to minimize discomfort, has smallest discomfort and largest time, while the flight scenario UA6, where all flights prefer to minimize time, has with smallest time and largest discomfort.

However, Figure 5.16(b) shows that the LCS cannot help flights to avoid all the bad weather areas as the algorithms using GA can do. This happens in all weather scenarios with bad weather cells.

Figures 5.16(a) and 5.16(b) show that the time and discomfort orders follow "*the orders of 5*". The discomfort order for the 7 utility allocations in Figure 5.16(b)



(a) Minimum objective through generations; Flights are given cruising altitudes and prefer to minimize time; Each line presents the results for one seed.



(b) Minimum objective through generations; Flights are given cruising altitudes and prefer to minimize discomfort; Each line presents the results for one seed.

Figure 5.15: Minimum objective through generations. Flights are given cruising altitudes and prefer to minimize discomfort or time

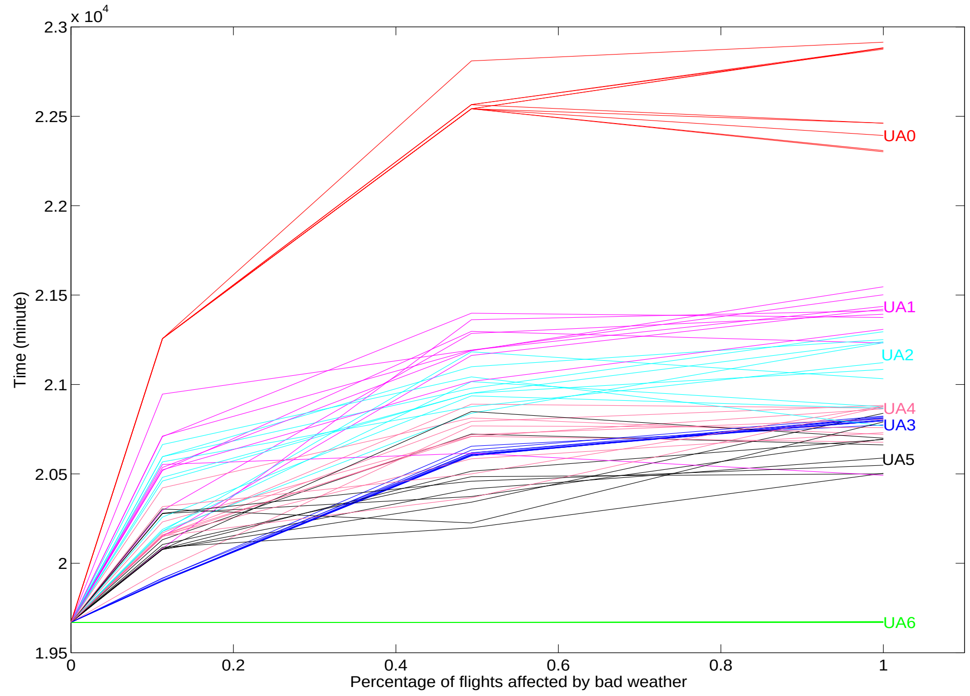
is the same to that in Table 5.3. The time order for the 7 utility allocations in Figure 5.16(a) is similar to that in Table 5.4 with a bit overlap between UA3 and UA5.

c) **An example classifier**

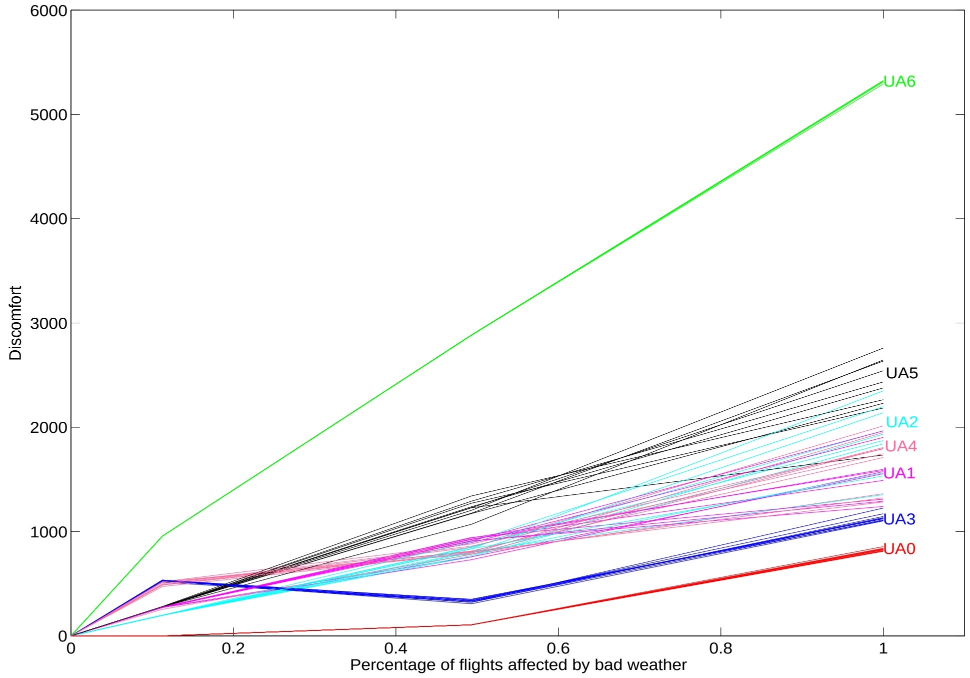
Table 5.12 presents the best classifier for the bad weather scenario affecting 50% of flights and the flight scenario UA2. The action of flying straight is used the most. Specifically the action of flying straight is used 776 times, while those for turning left 30, 20, and 10 (deg) and turning right 10, 20, and 30 (deg) are used 459, 407, 12, 28, 21, and 94 times respectively in the classifier.

From the classifier we can see a rule using a tactical maneuver to avoid unfavorable weather condition. The rule is "IF U = time AND BWL = none AND RWD = head THEN turn right 10". This means if an aircraft prefers to minimize time, there is no bad weather, and the wind is head wind, the aircraft will turn right to avoid the head wind. The other rules are for strategic maneuvers. By following the rules the aircraft will reach favorable weather and wind conditions and can then apply straight flying segments to their destination. This can help to achieve the global objective.

Table 5.13 summarizes the frequency of using commands: Turn ± 10, Turn ± 20, and Turn ± 30 (deg), for flights with comfort, neutral, and time preferences. It can be seen that flights preferring to minimize discomfort tend to make the largest turns (of 30 (deg)), to avoid bad weather areas. The flights preferring to minimize time tend to make small turns (of 10 (deg)), so that they can take the advantage of favorable wind and also avoid making large deviations from direct routes (which are the shortest). The flights with neutral preference tend to choose the medium turn (of 20 (deg)) so they can avoid bad weather areas while simultaneously minimizing time traveled.



(a) Time in different flight and weather scenarios; LCS method and GAP problem.



(b) Discomfort in different flight and weather scenarios; LCS method and GAP problem.

Figure 5.16: Time and discomfort in different flight and weather scenarios; LCS method and GAP problem.

Rule ID	Usage	Content
0	5	IF U = time AND BWL = none AND RWD = none THEN turn left 10
1	1	IF U = neutral AND BWL = med-low AND RWD = cross up THEN turn left 30
2	0	IF U = neutral AND BWL = medium AND RWD = cross up THEN turn left 20
4	0	IF U = comfort AND BWL = med-high AND RWD = none THEN fly straight
5	0	IF U = comfort AND BWL = med-low AND RWD = none THEN turn right 20
6	0	IF U = neutral AND BWL = low AND RWD = tail THEN turn right 30
7	28	IF U = time AND BWL = none AND RWD = head THEN turn right 10
9	0	IF U = neutral AND BWL = low AND RWD = head THEN turn right 20
10	1	IF U = time AND BWL = low AND RWD = tail THEN turn right 20
11	16	IF U = neutral AND BWL = none AND RWD = head THEN turn right 20
14	0	IF U = time AND BWL = med-high AND RWD = none THEN turn left 20
15	0	IF U = neutral AND BWL = high AND RWD = cross down THEN fly straight
16	0	IF U = time AND BWL = med-high AND RWD = tail THEN turn left 20
18	407	IF U = neutral AND BWL = none AND RWD = cross up THEN turn left 20
19	457	IF U = comfort AND BWL = none AND RWD = cross up THEN turn left 30
20	0	IF U = neutral AND BWL = med-low AND RWD = head THEN turn right 20
22	0	IF U = neutral AND BWL = med-high AND RWD = cross up THEN fly straight
24	0	IF U = comfort AND BWL = medium AND RWD = cross down THEN turn left 30
25	4	IF U = time AND BWL = high AND RWD = cross up THEN turn right 30
26	4	IF U = time AND BWL = med-high AND RWD = cross up THEN turn right 20
28	0	IF U = comfort AND BWL = medium AND RWD = head THEN turn right 10
29	1	IF U = comfort AND BWL = low AND RWD = cross up THEN turn left 30
30	0	IF U = neutral AND BWL = low AND RWD = none THEN turn left 20
31	0	IF U = time AND BWL = medium AND RWD = tail THEN turn left 20
33	90	IF U = comfort AND BWL = none AND RWD = tail THEN turn right 30
34	0	IF U = neutral AND BWL = high AND RWD = tail THEN turn right 30
36	0	IF U = neutral AND BWL = medium AND RWD = cross down THEN turn right 20
37	0	IF U = time AND BWL = med-low AND RWD = cross down THEN turn right 30
38	0	IF U = comfort AND BWL = med-low AND RWD = head THEN fly straight
39	0	IF U = comfort AND BWL = med-high AND RWD = cross down THEN turn right 30
40	0	IF U = comfort AND BWL = high AND RWD = head THEN turn right 10
41	0	IF U = time AND BWL = medium AND RWD = cross up THEN fly straight
42	0	IF U = time AND BWL = med-high AND RWD = head THEN turn left 30
44	0	IF U = comfort AND BWL = med-low AND RWD = cross down THEN turn left 20
45	7	IF U = neutral AND BWL = none AND RWD = none THEN turn left 10
47	0	IF U = comfort AND BWL = low AND RWD = head THEN fly straight
48	0	IF U = time AND BWL = high AND RWD = tail THEN turn left 10
50	776	fly straight

Table 5.12: An example of classifier for flights given cruising altitudes

	Turn ±10	Turn ±20	Turn ±30
Comfort	0.00%	0.00%	100.00%
Neutral	1.62%	98.14%	0.23%
Time	78.57%	11.90%	9.52%

Table 5.13: Frequency of using commands for flights given cruising altitudes

5.4.4 Conclusion

The designed LCS for the GRP and GAP problems successfully discovered rules in all runs to optimise the system performance. LCS can help to optimise both discomfort and time.

Although LCS cannot provide solutions as good as "black box" algorithms using GA, its solution is transparent to users, and reusable. LCS is also able to discover rules which can be applied to operations in air traffic control. Some of these rules may be new to ATC. We found a number of rules for tactical maneuver from the best classifiers. These rules agree with what happens in reality.

In addition, the discovered classifiers can be used to validate or generalize the results from "black box" algorithms. By using GA, we can find the best 3-D route for flights, but we may not have a general idea about the 3-D routes. LCS can help us to have a general view about the routes through the rules in the best classifier. For example, when we find UPR routes in no-wind and no-bad weather conditions by the algorithms using GA, it is difficult for us to have a general view of the best route. However LCS algorithm can find rules which can provide us with a better view. For example, two rules found here are "IF U = time AND BWL = none and RWD = none THEN climb" for flights given routes and "IF U = time AND BWL = none and RWD = none THEN fly straight" for flights given cruising altitudes. From these rules we can easily understand that in the given weather condition, flights which are given routes will climb as high as possible then cruise to TOD and continuously descend to destinations, and flights that are given cruising altitudes will fly their great circle route from origins to destinations.

5.5 Comparison of black and white box approaches

In general, the running times of GA (the black box approach) and LCS (the white box approach) are similar. GA can provide better UPR routes than LCS. However there is no explanation for UPR routes provided by GA, while LCS provides a set of ordered rules that are transparent to users and reusable. The departure time adjustments to resolve conflicts among UPR flights themselves and with non-UPR flights in the two methods are also almost the same. Details of comparison are presented in following sections.

5.5.1 Running Time

The experiments were run on HP Workstation Z400. The CPU type is Intel Xeon Dual-Core W3505 with the speed of 2.53 GHz.

Table 5.14 presents the running times to generate 142 3-D UPR routes by 2 methods using GA, and LCS for the two problems: where flights are given routes (GRP) and cruising altitudes (GAP). Time was measured for the weather scenario affecting 50% of UPR flights, and the flight scenario with the same user preference allocation for the 3 preferences (time, "neutral", and comfort).

All methods are designed to run in parallel. The running times are presented when the methods run in 2 computation nodes. The programs corresponding to the methods can run faster if more computation nodes are provided. For methods using GA, when a computation node is free, it will be used to find a 3-D UPR-route for a flight that hasn't yet got a UPR route. For methods using LCS, when a computation node is free, it will be used to evaluate a classifier in the population of classifiers in the current generation. Here, I use combined parallel work-sharing constructs [137] to assign these independent works to computation nodes. Particularly, the parallel for construct in OpenMP [137] is used.

The table shows that solving the GRP problem is a bit faster than solving the GAP problem. The running times of GA and LCS are similar. In terms of practical feasibility, the running times of GA and LCS are both around 125 minutes. If 8 computation nodes are available, which is easily feasible with current technology, both GA and LCS will take about 30 minutes. These running times are acceptable in UPR strategic planning.

In real time mode or in a decentralized routing environment (where the flight

	GA	LCS
GRP	123.00	121.90
GAP	136.84	133.72

Table 5.14: Running Times (minutes) for combinations of methods and problems.

crew can choose the flight route without or with minimal support from the ground), a UPR application is installed in an aircraft so that pilot can use it to choose their preferred route. The methods now need to find UPR for only one flight, so GA will take only about 2 minutes. However GA can be designed to run in parallel while it find UPR for one flight, along the lines of the design of LCS, where an individual in a population is assigned to a computation node for evaluation when the node is free. With this design, GA can find the UPR route for a flight in about 13 seconds by using 8 computation nodes in our experimental environment.

After having found all UPR routes for UPR flights, conflicts between these routes, and conflicts with non-UPR routes, are detected and resolved. The running time for conflict detection and resolution for UPR flights with each other and with non-UPR flights takes only about 30 seconds.

5.5.2 Quality of UPR routes

Table 5.15 presents a comparison of UPR methods with each other and with the original flight plans in two flight scenarios. The first flight scenario is UA6 with all flights preferring to minimize time. The second is UA0 with all flights preferring to minimize discomfort. The bad weather scenario that affects all the UPR flights is used. "M & P" is the combinations of a method which uses GA, or LCS and a problem which is GRP, or GAP. "FS" is flight scenarios. "Pref Time" and "Pref DC" correspond to flight scenarios UA6 and UA0. "Value" is the values of time, fuel, or discomfort objective depending on which column ("Time", "Fuel", and "Discomfort") it belongs to. "%Diff" is the difference in percentage between the objectives provided by UPR methods and the ones provided by the original flight plans.

		Time		Fuel		Discomfort	
M & P	FS	Value	%Diff	Value	%Diff	Value	%Diff
F Plan	F Plan	20116.9	0.00%	986958.1	0.00%	4558.7	0.00%
GA - GRP	Pref Time	18620.2	-7.44%	897415.8	-9.07%	3042.4	-33.26%
	Pref DC	18845.2	-6.32%	870110.2	-11.84%	14.9	-99.67%
GA - GAP	Pref Time	19651.6	-2.31%	968259.9	-1.89%	4928.6	8.11%
	Pref DC	19853.5	-1.31%	971931.1	-1.52%	22.2	-99.51%
LCS - GRP	Pref Time	19349.0	-3.82%	976316.4	-1.08%	4973.9	9.11%
	Pref DC	21252.8	5.65%	892173.6	-9.60%	2534.2	-44.41%
LCS - GAP	Pref Time	19670.1	-2.22%	965300.3	-2.19%	5320.9	16.72%
	Pref DC	22392.6	11.31%	1086769.0	10.11%	855.5	-81.23%

Table 5.15: Time, Fuel, and Discomfort of the original flight plans and UPR flight routes in different flight scenarios.

In general, GA can provide solutions with higher quality than LCS. For each flight scenario and each combination of a problem and a method, all the objective values of solutions provided by LCS are higher (worse) than those of solutions provided by GA.

The table also shows that when the flight scenario is “Pref DC”, GA can reduce the discomfort objective by more than 95% in comparison with the original flight plans, while LCS can reduce discomfort by about 44% and 81% for problems GRP and GAP respectively; when the flight scenario is “Pref Time”, the time objective can be reduced by more than 2% in every problem-method combination and the fuel burn can be reduced by about 2% except for the combination of “LCS - GRP” with a 1.08% reduction. The reduction of 2% fuel burn results in a fuel saving of about 150 kg per flight in average. This reduction is normal with the normal wind conditions as presented in other studies (e.g in [98]).

It can be seen that in the problem where flights are given cruising altitudes (GAP), when the time travelled is high the fuel burn is also high. This is because fuel flow for an aircraft is the same at an altitude and the cruising phase in the given constant cruising altitude accounts for most of the route of the aircraft.

However, this does not apply in the problem where flights are given routes (GRP). In some cases when the time travelled is high, the fuel is low because of the trade off between fuel burn and ground speed while an aircraft changes its altitude.

		Dept Time Deviation (min)	Conflict with UPR	Conflict with Non-UPR
GA	GRP	63	40	36
	GAP	65	41	37
LCS	GRP	64	39	38
	GAP	63	39	37

Table 5.16: Departure Time Deviation and Conflicts between UPR flights with UPR and Non-UPR flights.

5.5.3 Conflict Detection and Resolution

After UPR flights have UPR routes, KB3D is applied to detect conflicts between these routes with each other and with Non-UPR routes. Conflicts are then resolved by departure time adjustment.

Depending on the flight and weather scenarios, the total deviation of departure time from the original ones across all 142 UPR flights is from 63 to 65 minutes. The maximum departure time deviation of a flight is 2 minutes. On average a flight only needs to adjust its departure times by less than 1 minute. Table 5.16 presents the departure time deviation and the number of conflicts between UPR flights with UPR and Non-UPR flights. “Dept Time Deviation (min)” is the total departure time deviation of all the flights in minutes; “UPR conflict” is the number of conflicts between UPR flights with each other; “Non-UPR conflict” is the number of conflicts between UPR flights and Non-UPR flights; “GA” is the methods using Genetic Algorithms; “LCS” is the methods using Learning Classifier Systems; “GRP” is the problem to find UPR routes for flights given routes; “GAP” is the problem to find UPR routes for flights given cruising altitudes. The figures in the table are from the flight scenario with the utility allocation UA0 where all flights prefer to minimize discomfort; and the weather scenario affecting all the flights.

5.6 Conclusion

In this chapter, we propose methodologies to solve UPR problems by both black and white box approaches. The experiments show that both approaches are able to find good solutions. Though the solutions provided by the white box approaches are not as good as the black box approaches, their solutions are understandable to users. The running times and the departure time adjustments for conflict resolution of the both approaches are almost the same.

However both these heuristic approaches are quite time consuming. Depending on the computers available, they may not work well in real time or tactical route planning conditions. In addition, in the stage of transition from a fixed route structure to a free route structure, where the support from air traffic controllers are still needed, the user preferred routes provided by these approaches may be difficult for air traffic controllers to control as they do not relate to any structure. In the next chapter, we will propose models for generating dynamic networks. Then user preferred routes can be obtained from these networks. This approach can provide flexible and best routes for users, and also support air traffic controllers in controlling aircraft through the relation between the routes and the network structures (though they are dynamic).

Chapter 6

3-D User Preferred Routes using Dynamic Network of Control Points

6.1 Overview

In this chapter, we propose methods to solve both GRP and GAP problems, based on dynamically generated networks of control points. These methods are called NetM methods. In these methods, algorithms to generate a network of control points for a flight, which is given a route or a cruising altitude in GRP and GAP problems respectively, are proposed. The validated aircraft performance is used to determine nodes (control points) and weights (Distance, Time, Discomfort, Fuel) for a network. In order to find user preferred routes for all the flights, a network is generated for each. A network can then be used to find the 3-D user preferred route for a flight efficiently by a shortest path algorithm. The conflicts between user preferred routes are then detected and resolved by the model, presented in Section 3.8.

There has been research constructing networks for finding optimal aircraft trajectory/route [41, 133, 128, 20, 202, 201, 111]. There are many different ways to construct a network. Some divide the airspace to a grid of cells and each cell is considered as a node [133]. Other use available waypoints to construct a network [41]. However, the network is usually constructed in 2-D [41, 133] or does not use real aircraft performance [41, 133, 128] or does not consider wind factor in calculating weights of time, fuel and so on.

In this chapter we propose innovative network models and algorithms to generate these dynamic networks of 3-D control points using real aircraft performance and real wind data. The 3-D network can be also considered as a 4-D network as the weight of time travelled is determined in links of the network.

The advantage of the generated network is that a network is generated for a combination of an origin, a destination, an aircraft type, and a weather environment, but it can be used to find UPR routes for all the flights having this combination of information. For example, flights with different user preferences or different departure times can use the same network to find their optimal routes. A network is generated in one time but it can be used in many times. A network generated is successful to find UPR routes.

The remainder of this chapter is organized as follows. Sections 6.2 and 6.3 present network models, and algorithms for generating a network for a flight given a 2-D route or given a cruising altitude, and algorithm for finding UPR route based on the generated network. Section 6.4 proposes methods to optimise aircraft route both vertically and horizontally, and analyses the difference among methods using network models. Section 6.5 compares methods using networks with those using GA and LCS, and proposes a systematic use of these methods in air traffic management operation. Section 6.6 analyses the environmental impact of UPR routes using different methods. Conclusions are presented in Section 6.7.

6.2 3-D User Preferred Routes using Dynamic Network of Control Points for a flight given a 2-D route

In this section, NetM for GRP problem is designed. We propose an algorithm to generate a network of control points for a flight (in GRP problem), which is given a 2-D (latitude, longitude) route and is allowed to change its altitude during cruise phase. Then Dijkstra's algorithm is used to find a 3-D (latitude, longitude, altitude) user preferred route for the flight.

6.2.1 Network Generation

The problem is given a flight with specified aircraft performance and a 2-D (latitude, longitude) route. The aircraft performance is given in BADA data [135]. The route is a series of 2-D (latitude, longitude) waypoints from origin to destination. The task is to generate a network of control points for the flight.

We define that 2-D (latitude, longitude) grid waypoints are points belonging to great circle routes between two consecutive waypoints of the 2-D route. The first grid waypoint of a great circle route between two waypoints is the first waypoint. Each succeeding grid waypoint is a constant distance (a predefined value) from the previous one. The last grid waypoint is the second waypoint.

The nodes of a network can belong to one of six categories:

1. Grid node: Grid nodes are 3-D (latitude, longitude, altitude) points. The latitude and longitude of a grid node are the ones of a grid waypoint and its altitude is one of the standard flight levels.
2. Climb node: Climb nodes are ending points where the aircraft climbs from grid nodes in a flight level to the next upper flight level.
3. Descent node: Descent nodes are ending points where the aircraft descends from grid nodes in a flight level to the next lower flight level.
4. TOC (Top Of Climb) node: TOC nodes are ending points where the aircraft climbs from origin to flight levels.

5. TOD (Top Of Descent) node: TOD nodes are starting points where the aircraft descends to the destination airport.
6. Origin and Destination nodes: They are origin and destination airports.

The links of a network are divided to 3 groups: climb links, cruise links, and descent links. The weights (Time, Distance, Fuel, Emissions) are calculated for each link.

1. Climb links
   - Climb link from the origin to a TOC node: This link is created between the origin and a TOC node the aircraft climbs from the origin to.
   - Climb link from a grid node to a climb node: This link is created between a grid node and the climb node the aircraft climbs from the grid node to.
2. Cruise links
   - Cruise link from a grid node to a grid node: This link is created between two consecutive nodes in a flight level.
   - Cruise link from a grid node to a TOD node: This link is created between a grid node and the TOD node next to the grid node in the same flight level
   - Cruise link from climb node to grid node: This link is created between a climb node and the grid node next to the climb node in the same flight level
   - Cruise link from a descent node to grid node: This link is created between a descent node and the grid node next to the descent node in the same flight level.
   - Cruise link from a TOC node to a grid node: This link is created between a TOC node and the grid node next to the TOC node in the same flight level.
   - Cruise link from a climb node to a TOD node: This link is created between a climb node and the TOD node next to the climb node in the same flight level
   - Cruise link from a descent node to a TOD node: This link is created between a descent node and the TOD node next to the descent node in the same flight level
3. Descent links
   - Descent link from a grid node to a descent node: This link is created between a grid node and the descent node the aircraft descends from the grid node to.

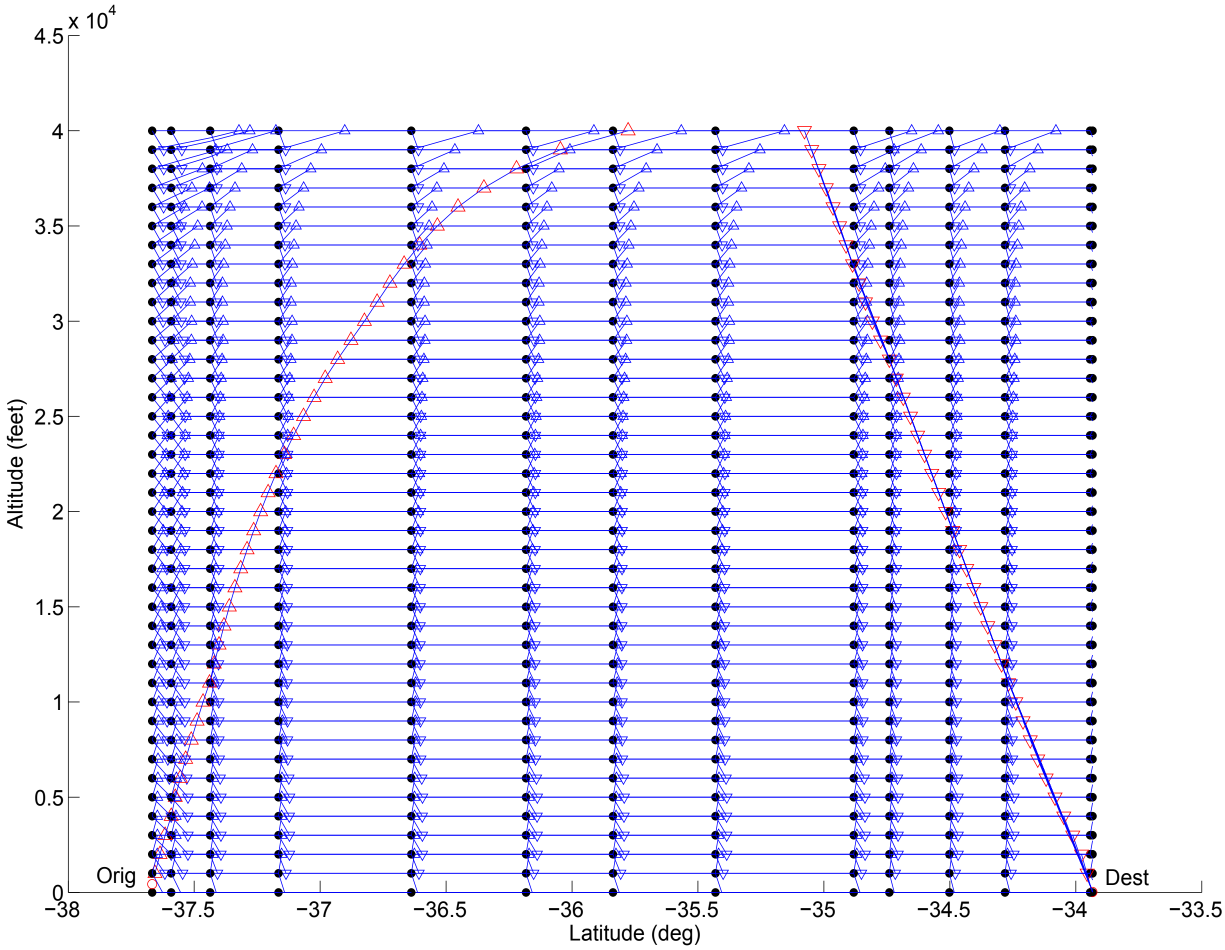


Figure 6.1: Network of control points for a flight.

- Descent link from a TOD node to the destination: This link is created between a TOD node and the destination node.

Figure 6.1 is one example network for a flight from Melbourne to Sydney. The aircraft type of the flight is B764. In the figure "Orig" and "Dest" are the origin and destination airports. Origin and destination nodes are presented by symbol ∘ in red; Grid nodes by • in black; Climb nodes by ▲ in blue; Descent nodes by ▼ in blue; TOC by △ in red; TOD by ▽ in red.

A network is represented by a list of nodes (LNS). Each node has node index, and the list of links to its next nodes and the list of its previous nodes which are used to track the 3-D route. Each link includes the index of the starting node and the next node, weights (Distance, Time, Discomfort, Fuel), and a list of sub-links which are simulated moving steps of the flight through the link. Firstly origin and destination nodes (which are the origin and destination airports) are created and added into LNS.

6.2.1.1 Grid Nodes

Algorithm 8 is to generate grid nodes. In order to generate grid nodes we predefine the width between two consecutive grid nodes (GW) and flight levels. Grid nodes are created for each waypoint of the route, and between every two consecutive waypoints of the flight route in different flight levels. The distance between a grid node to its next grid node is GW, except for the case the next node is also a waypoint in the route. In this case, the distance may be shorter than GW. A link is created between every two consecutive grid nodes. The weights and the list of sub-links of a link are determined by the simulation of the aircraft cruising from one point to another point as presented in Section 3.7.10.

Algorithm 8 Creating grid nodes

n is the number of waypoints of the route
for each flight level **do**
  for each $waypoint_i$ from the first waypoint to $(n-1)^{th}$ waypoint of the route **do**
    $nextwp$ = the next waypoint of $waypoint_i$.
    Creating a new grid node ($newnode$) whose latitude and longitude are the ones of $waypoint_i$ and altitude is the flight level. Adding the new grid node to LNS.
    $currentnode = newnode$
    while Distance from $currentnode$ to the $nextwp$ is more than GW **do**
      Creating a new grid node ($newnode$) which is horizontally between $currentnode$ and $nextwp$ and the horizontal distance between $newnode$ and $currentnode$ is GW, and its altitude is the flight level. Then adding $newnode$ to LNS. Adding a link from $currentnode$ to $newnode$ to the list of links of $currentnode$. This link is a cruise link. The ways to determine the weights (Distance, Time, Discomfort, Fuel) and the list of sub-links of the link will be presented in Section 3.7.10.
      $currentnode = newnode$
    end while
    Creating a new grid node ($newnode$) whose latitude and longitude are the ones of $nextwp$ and altitude is the flight level. Adding the new grid node to LNS. Adding a link from $currentnode$ to $newnode$ to the list of links of $currentnode$.
  end for
end for

6.2.1.2 Climb Nodes

Climb nodes and their links are determined by climbing an aircraft from a grid node to the flight level above the one of the grid node. Figure 6.2 presents climb

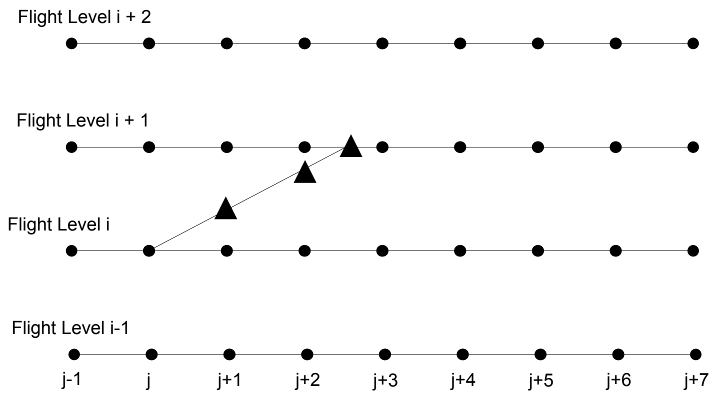


Figure 6.2: Aircraft climbs from a grid node to the upper flight level.

nodes (▲) and their links when the aircraft climbs from grid node j in flight level i (FL_i) to the flight level $i+1$ (FL_{i+1}).

The determination of the climb point, the weights, and the list of sub-links of the climb link is presented in Section 3.7.1. If the climb point coincides with a grid node, the grid node is also the climb node. Otherwise a climb node is created at the climb point. A link between a grid node and the climb node is created. If there is a grid node next to the climb node, a cruise link from the climb node to the grid node is also created. The weights and list of sub-links of the cruise link are determined in Section 3.7.10.

In Figure 6.2, the aircraft follows 2-D (latitude, longitude) route of 4 consecutive grid nodes j, $j+1$, $j+2$, and $j+3$ to climb from grid node j in FL_i to FL_{i+1}. The figure presents the points where aircraft passes grid waypoints and the final climb point by symbol ▲. A climb link from the grid node to the climb node is created. In addition a cruise link between the climb node and the grid node $j+3$ in FL_{i+1} is created.

6.2.1.3 Descent Nodes

Descent nodes and their links are determined by descending an aircraft from a grid node to the flight level below the one of the grid node. Figure 6.3 presents descent nodes (▼) and their links when the aircraft descends from grid node j in

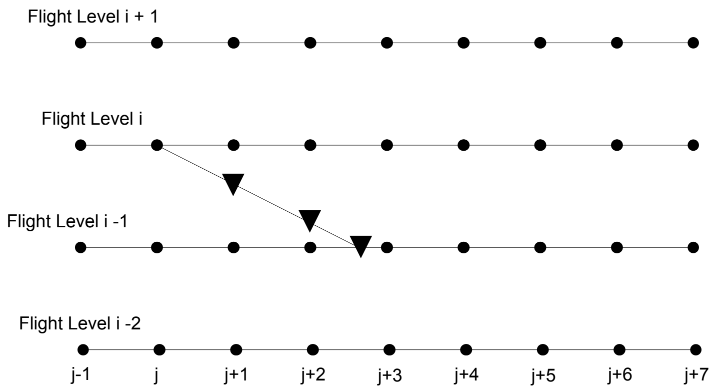


Figure 6.3: Aircraft descends from a grid node to the lower flight level

flight level i (FL_i) to the flight level $i-1$ (FL_{i-1}).

The descent point, the weights, and the list of sub-links of the descent link are determined as presented in Section 3.7.4. If the descent point coincides with a grid node, the grid node is also the descent node. Otherwise a descent node is created at the descent point. A link between a grid node and the descent node is created. If there is a grid node next to the descent node, a cruise link from the descent node to the grid node is also created. The weights and the list of sub-links of the cruise link are determined as in Section 3.7.10.

In Figure 6.3, the aircraft follows 2-D (latitude, longitude) route of 4 consecutive grid nodes j, $j+1$, $j+2$, and $j+3$ to descend from grid node j in FL_i to FL_{i-1}. The figure presents the points where aircraft passes grid waypoints and the final climb point by symbol ▼. A descent link from the grid node to the descent node is created. In addition a cruise link between the descent node and the grid node $j+3$ in FL_{i+1} is created.

6.2.1.4 TOC Nodes

A TOC node is determined in the same way as a climb node. The only difference is that instead of climbing from a grid node to the next flight level the aircraft climbs from the origin to a certain flight level. The final climb node is the TOC node as in Figure 6.4. In Figure 6.4, the aircraft follows 2-D (latitude, longitude) routes of

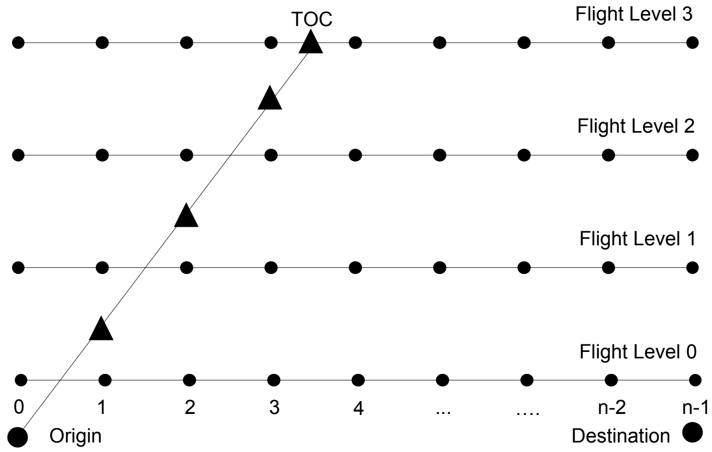


Figure 6.4: Aircraft climbs from the origin airport to the Top Of Climb (TOC) of a given flight level.

5 consecutive nodes (the origin node and 4 grid nodes 1, 2, 3, 4) to climb from the origin j to FL_3 step by step. At each step a climb node and a climb link from the the current node to the climb node are created. In addition a cruise link between the climb node and grid node 4 in FL_3 is created at the final step.

6.2.1.5 TOD Nodes

A TOD node is determined by tracing backward from the destination to a flight level. Figure 6.5 presents the example of a TOD node and descent nodes (▼) and links when tracing backward from the destination to flight level i (FL_i), where i is 3.

The TOD point, the weights, and the list of sub-links of the descent link are determined as presented in Section 3.7.7. If the TOD point coincides with a grid node, the grid node is also the TOD node. Otherwise a TOD node is created at the TOD point. A link between a grid node and the TOD node is created. If there is a grid node previous to the TOD node, a cruise link from the grid node to the TOD node is also created. The weights and the list of sub-links of the cruise link are determined as in Section 3.7.10.

In Figure 6.5, the aircraft follows the 2-D (latitude, longitude) route of 5 consecutive nodes including origin node and 4 grid nodes $n-2$, $n-3$, $n-4$, $n-5$ to

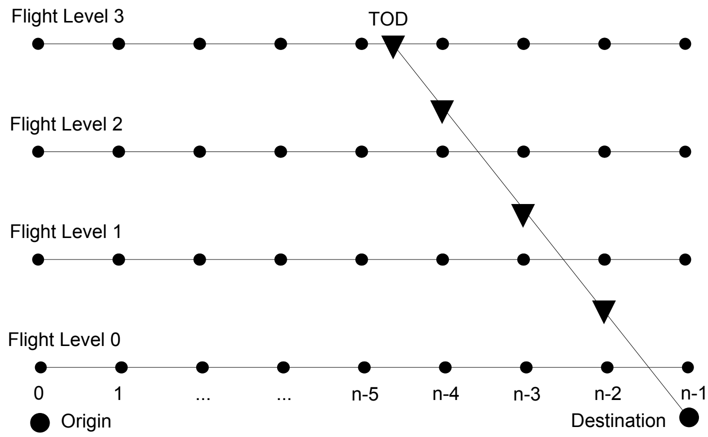


Figure 6.5: Aircraft moves backward from the destination airport to the Top Of Descent of a given flight level.

move backward from origin j to FL_3. The figure presents the points where aircraft passes grid waypoints and the TOD node by symbol ▼. A descent link from TOD node to the destination node is created. In addition a cruise link between the grid node $n-5$ and the TOD node in FL_3 is created.

6.2.2 Finding the 3-D user preferred route for a flight given 2-D route

After having the network for a flight where the distance, time, discomfort, and fuel weights of every link are calculated, a combined weight of time and discomfort is calculated by Equation 6.1 where u is the utility of the flight.

$$CombineWeight = u * time + (1-u) * discomfort \tag{6.1}$$

Dijkstra's algorithm is used to find the shortest path from the origin to the destination based on the network, where the weight of a link is the combined weight. Each node of the network has properties: the shortest distance from the origin to the node (*dist*); the list of links to next nodes (*nextlinks*); the previous node of the node in the shortest path (*previous*); a property determining if the dist of the node is permanently determined (*perm*). Each link of the network has properties: starting and ending nodes; weights of the link: time, fuel, discomfort; the combined weight: dist which is considered as the distance from the stating node to the ending node of the link. An object (a node or a link) accesses its properties by operator dot.

We adopt Dijktra's algorithm [180] to resolve the problem as in Algorithm 9. The original algorithm is modified. A set of nodes Q which stores nodes whose dist is assigned but not permanently determined is maintained, while the original algorithm maintains a set of all nodes whose dist is not permanently determined. This modification helps to speed up the algorithm because Q stores only a few nodes so the node with the smallest dist in Q can be found faster.

Algorithm 9 Algorithm for finding UPR route

```
for each node v in the network do
    v.dist= infinity {Unknown distance function from origin to v}.
    v.previous = undefined {Previous node in optimal path from origin}.
    v.perm = false
end for
origin.dist = 0 {Distance from origin to origin}.
Q = origin {the set of nodes whose dist is assigned but not permanently determined}.
while Q is not empty do
    u = node in Q with smallest distance;
    if u = destination then
        break
    end if
    for each link l of node u do
        {u and v are the starting and ending nodes of link l}.
        if v.perm = false then
            alt = u.dist + l.dist
            if v.dist = infinity then
                v.dist = alt
                v.previous = u
                insert v into Q
            else
                v.dist = alt
            end if
        end if
    end for
    u.perm = true
    remove u from Q
end while
P = empty {set of nodes of the shortest path from the origin to the destination}
{insert nodes from the destination to the origin into P}
curnode = the destination
while P.previous <> the origin do
    insert curnode into P as the first element.
    curnode = curnode.previous
end while
curnode = curnode.previous {curnode become the origin}
insert curnode into P as the first element
```

6.2.3 Parameter Settings

The following parameter settings are used:

- Flight levels are standard flight levels.
- The width between two consecutive grid node (GW) is 100 km.
- Distance of a sub-segment/ a simulated moving step is 1 km.

6.2.4 Results

6.2.4.1 Time and Discomfort Objectives of UPR flights given routes

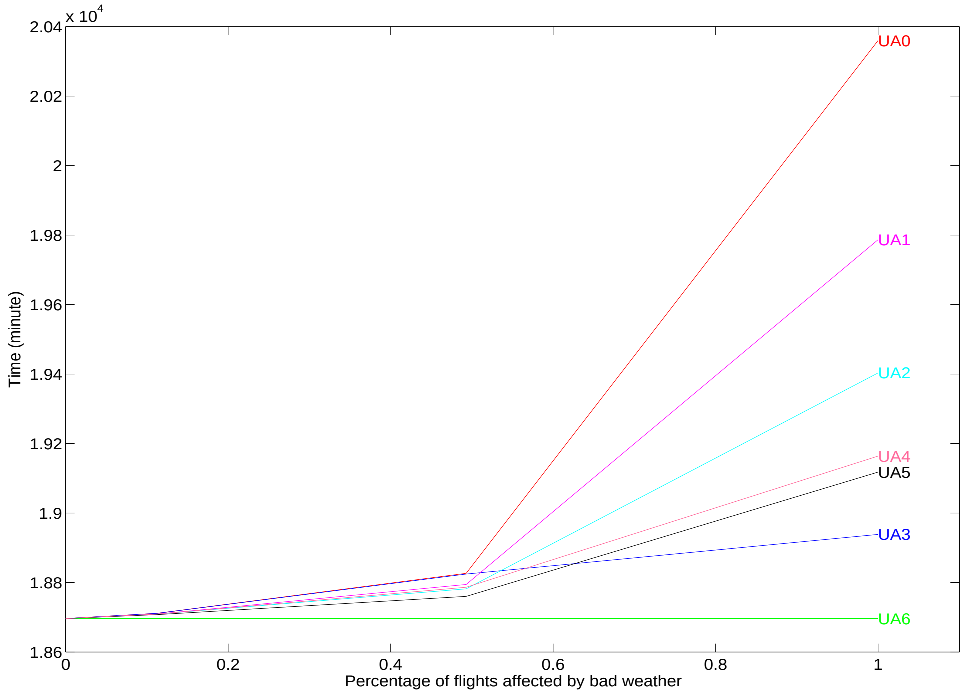
Figure 6.6(a) shows the time travelled by UPR flights with different flight utility allocations in different weather scenarios. A flight is given a route. The utility allocations from Table 3.3 can be UA0, UA1..., UA6 which present the percentage of flights preferring to minimize time, discomfort, or both time and discomfort with the same preference. The weather scenarios are presented by percentages of UPR flights affected by bad weather cells.

The figure shows that the algorithm is successful to find UPR routes. UPR flights with the utility allocation UA0, where all UPR flights prefer to minimize discomfort, travel the longest time; while those with utility allocation UA6, where all UPR flights prefer to minimize time, travel the shortest time. In the case where all UPR flights prefer to minimize time, the total time is not affected by bad weather scenarios as they all follow the most advantageous wind without bad weather consideration. In general the total times travelled of different flight utility allocations are directly proportional to the percentage of flights preferring to minimize discomfort.

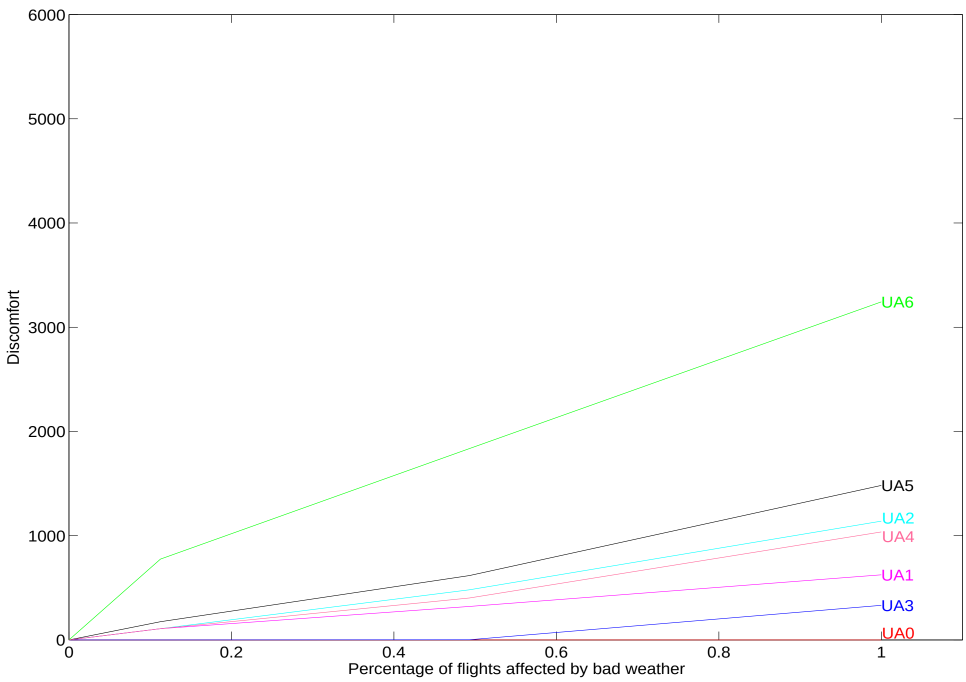
Figure 6.6(b) shows the discomfort of UPR flights with different flight utility allocations in different weather scenarios. The figure shows that UPR flights with the utility allocation UA6, where all UPR flights prefer to minimize time, are most affected by bad weather; while those with utility allocation UA0, where all UPR flights prefer to minimize discomfort, are not affected by bad weather. This shows that the algorithms can find UPR route for a flight so that the flight can avoid passing any area of bad weather when the preference of the flight is minimum discomfort. In general the total discomforts of different flight utility allocations are directly proportional to the percentage of flights preferring to minimize time.

6.2.4.2 Examples of UPR routes for flights given routes

Figure 6.10 is an example of two 3-D UPR routes of a flight which is given a route. One route prefers to minimize time and the other prefers to minimize discomfort. The 3-D route preferring to minimize time keeps its time optimal cruising altitude, while in the other the flight descends from its optimal cruising altitude to avoid passing the bad weather area. Bad weather cells are black cube with sizes of 1° x 1° x 5000 feet.

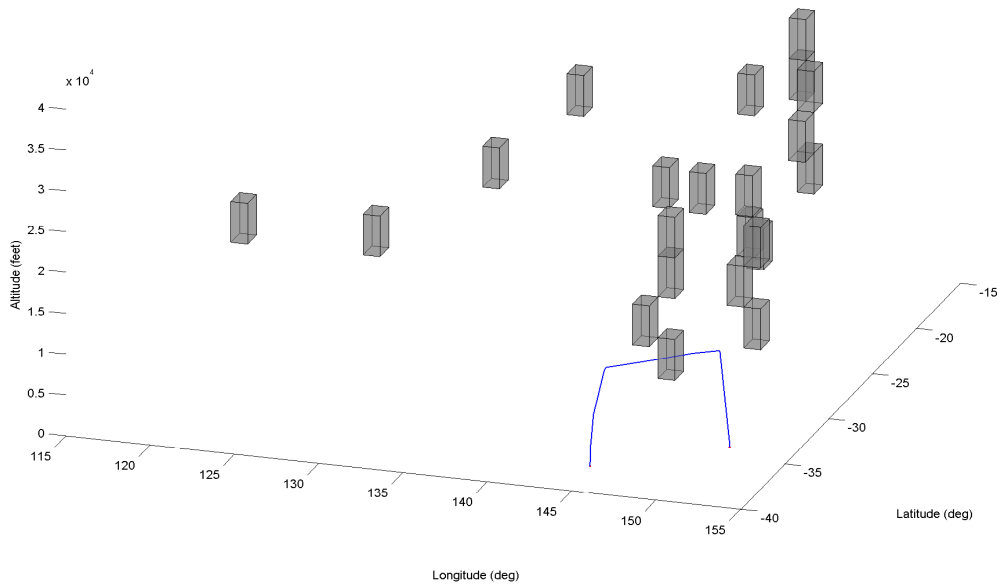


(a) Time in different flight and weather scenarios; Flights are given routes.

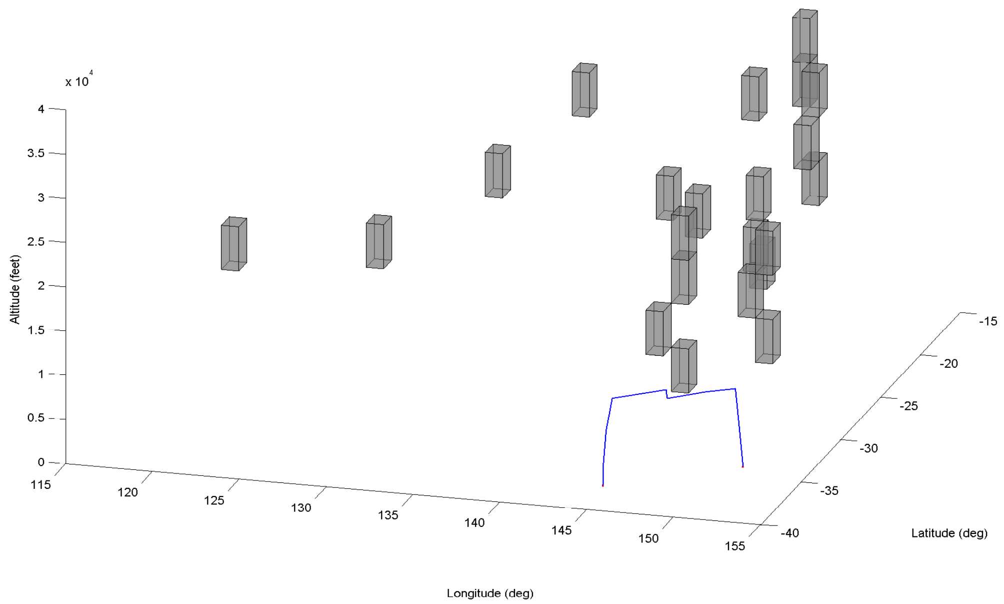


(b) Discomfort in different flight and weather scenarios; Flights are given routes.

Figure 6.6: Time and discomfort in different flight and weather scenarios; Flights are given routes



(a) 3-D route preferring to minimize time; The flight is given a route.



(b) 3-D route preferring to minimize discomfort; The flight is given a route.

Figure 6.7: Two 3-D routes preferring to minimize time and discomfort; The flights are given routes

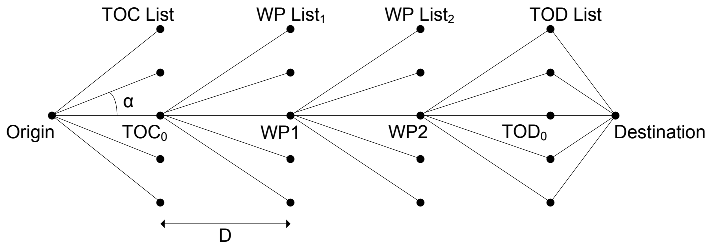


Figure 6.8: Discretized Grid for a flight with origin, destination and cruising altitude.

6.3 3-D User Preferred Routes using Dynamic Network of Control Points for a flight given a cruising altitude

In this section, NetM method for GAP problem is designed. We propose an algorithm to generate a network of control points for a flight (in GAP problem), which is given a cruising altitude and does not change altitude during cruise phase. Then Dijkstra's algorithm is used to find a 3-D (latitude, longitude, altitude) user preferred route for the flight.

6.3.1 Network Generation

In order to generate a network for a flight which is given a cruising altitude, firstly a list of TOC are generated by climbing aircraft up to the given cruising altitude via different directions as Figure 6.8. A direction is deviated from the initial bearing between the origin and the destination an angle of $n * \alpha$. α is a given angle as in Figure 6.8 and n is an integer number with constraint that $|n * \alpha| < 45^o$. A list of TOD are also generated by moving the aircraft backward from the destination to the given cruising altitude via different directions. Call the TOC and TOD whose deviation is $0°$ TOC_0 and TOD_0. A number of waypoints between TOC_0 and TOD_0 are generated. These waypoints divide the great circle route between TOC_0 and TOD_0 into a number of segments with the same distance D apart. A list of points is generated for each point from TOC_0 to the second last point before TOD_0 which is WP_1 in the example in Figure 6.8. The way to generate a list of points from a point is similar to generating the TOC list. The only difference is that instead of climbing aircraft to the given cruising altitude, the aircraft cruises from the point a distance of D via different directions. A network of control points is constructed by connecting a point from a previous list to all points in the next list. We can consider the origin is the first list and the destination is the last list with only one point.

6.3.2 Finding the 3-D user preferred route for a flight given a cruising altitude

After generating the network for a flight where the distance, time, discomfort, and fuel weights of every link are calculated, a combined weight of time and discomfort is calculated by Equation 6.1 where u is the utility of the flight. Then Dijkstra is used to determine the 3-D (latitude, longitude, altitude) user preferred route for the flight in the network with combined weight links. This works the same to finding 3-D UPR route for a flight given a route.

6.3.3 Parameter Settings

The following parameter settings are used:

- α (angle of a deviated unit) is 10°
- n (largest number of deviated units) is 4
- D (distance to divide the great circle route between TOC_0 and TOD_0) is 1/5 the great circle distance between origin and destination.
- Distance of a sub-segment (a simulated moving step) is 1 km.

6.3.4 Results

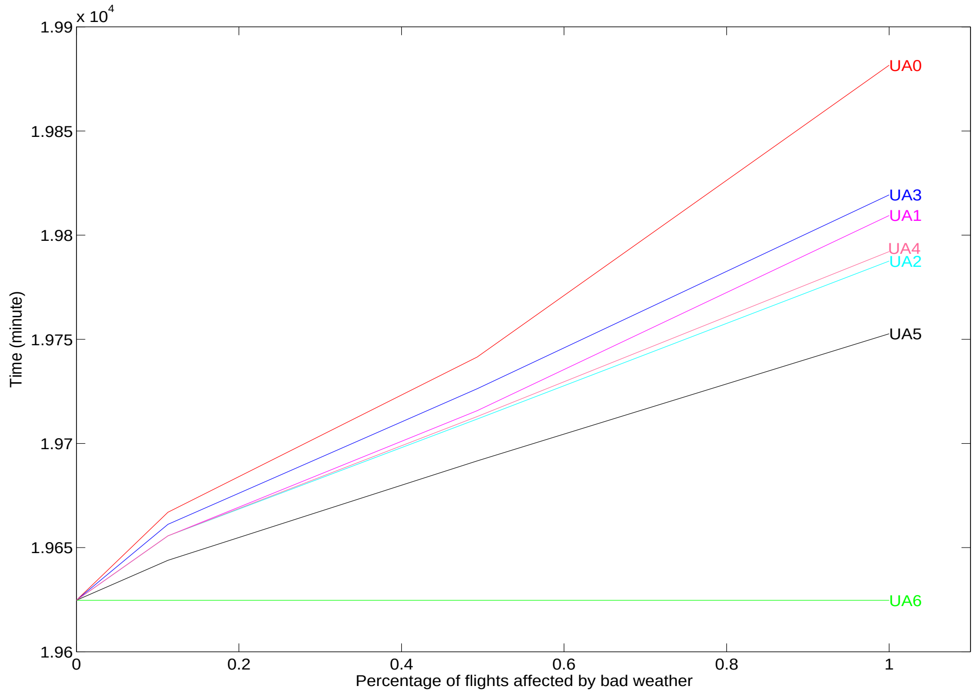
6.3.4.1 Time and Discomfort Objectives of UPR flights given cruising altitudes

Figure 6.9(a) shows the time travelled by UPR flights with different flight utility allocations in different weather scenarios. A flight is given cruising altitude. The figure shows that UPR flights with the utility allocation UA0, where all UPR flights prefer to minimize discomfort, travel the longest time; while those with utility allocation UA6, where all UPR flights prefer to minimize time, travel the shortest time and the total time is not affected by bad weather scenarios as they all follow the most advantageous wind. In general the total times travelled of different flight utility allocations are inversely proportional with the percentage of flights preferring to minimize time.

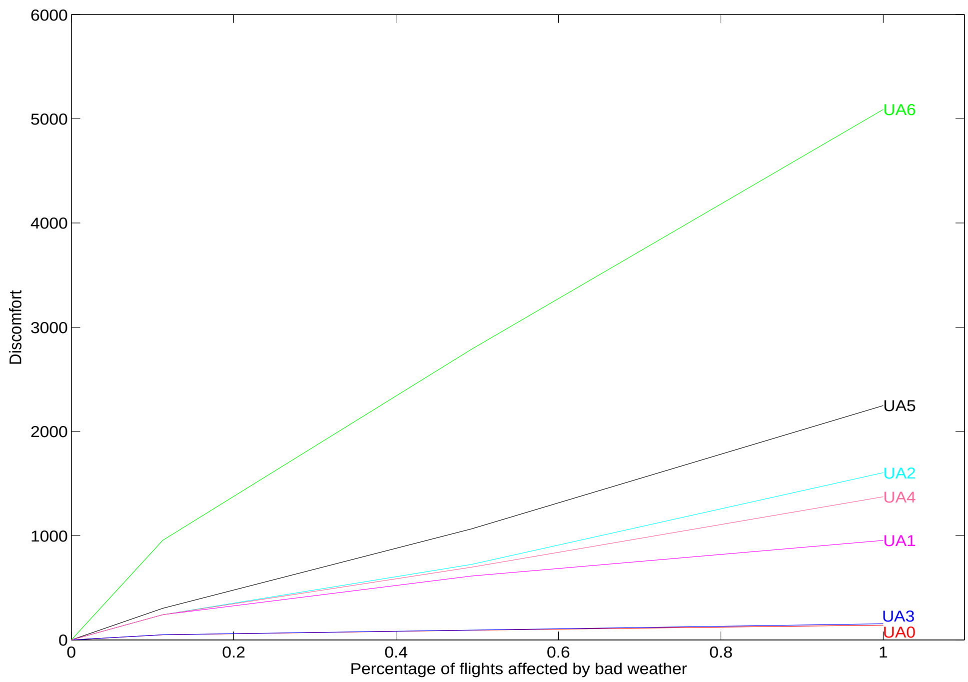
Figure 6.9(b) shows the discomfort of UPR flights with different flight utility allocations in different weather scenarios. A flight is given cruising altitude. The figure shows that UPR flights with the utility allocation UA6, where all UPR flights prefer to minimize time, are most affected by bad weather; while those with utility allocation UA0, where all UPR flights prefer to minimize discomfort, are almost not affected by bad weather. The discomfort of flights with utility allocation UA0 is approximate to 0. In general the total discomforts of different flight utility allocations are proportional to the percentage of flights preferring to minimize time.

6.3.4.2 Examples of UPR routes for flights given cruising altitudes

Figure 6.10 is an example of two 3-D UPR routes of a flight which is given a cruising altitude. One route prefers to minimize time and the other prefers to minimize discomfort. The 3-D route preferring to minimize time keeps its time optimal route, while in the other the flight makes a turn from its optimal route to avoid passing the bad weather area. Bad weather cells are black cube with sizes of 1° x 1° x 5000 feet.

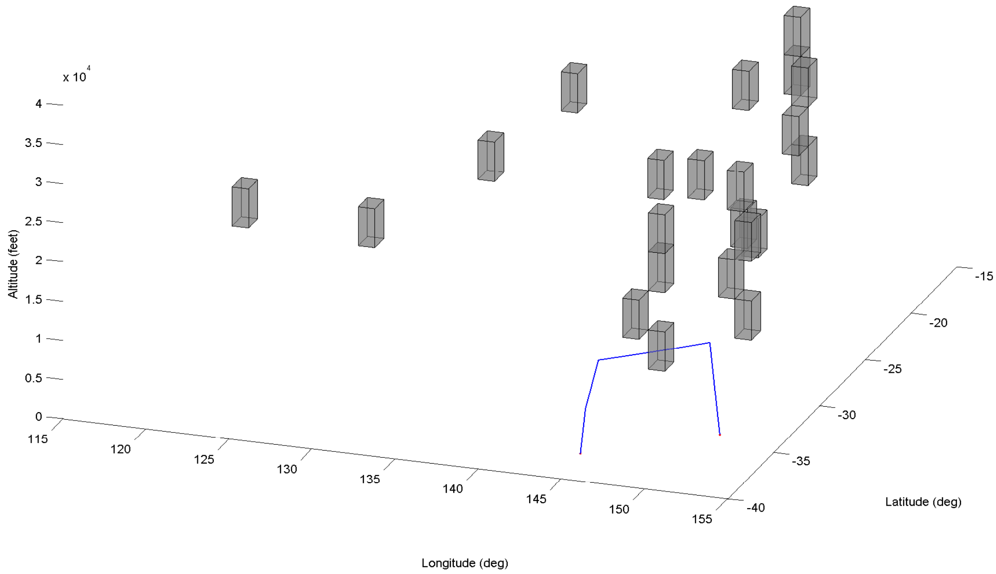


(a) Time in different flight and weather scenarios; Flights are given cruising altitudes.

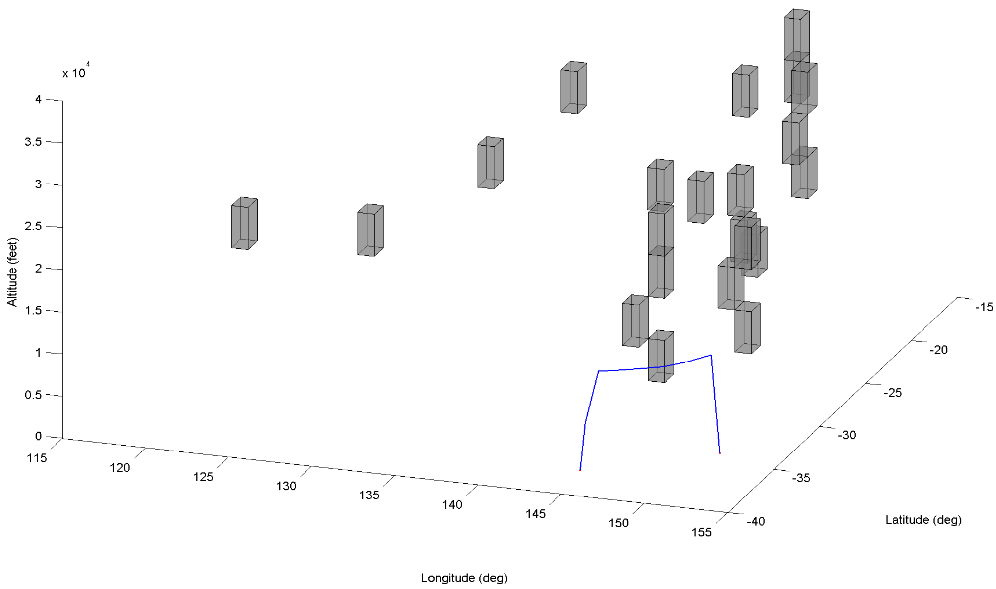


(b) Discomfort in different flight and weather scenarios; Flights are given cruising altitudes.

Figure 6.9: Time and discomfort in different flight and weather scenarios; Flights are given cruising altitudes.



(a) 3-D route preferring to minimize time; The flight is given cruising altitude.



(b) 3-D route preferring to minimize discomfort; The flight is given cruising altitude.

Figure 6.10: Two 3-D routes preferring to minimize time and discomfort; The flights are given cruising altitudes.

6.4 Horizontally and vertically optimised UPR routes using NetM

6.4.1 Proposed methodology

We will see in Section 6.5 that NetM is the most efficient method for UPR planning. However this method is so far only developed for optimising aircraft routes vertically or horizontally. In this section we propose to optimise UPR routes both horizontally and vertically. Figure 6.11 presents two approaches to resolve this problem. In Figure 6.11(a), the optimal cruising altitude of a UPR flight is first found by vertical optimisation as presented in Section 6.2 and then this cruising altitude is the input for horizontal optimisation as presented in Section 6.3. In Figure 6.11(b) the horizontal route of a UPR flight is first found by horizontal optimisation and then this horizontal route is the input for vertical optimisation.

6.4.2 Difference between methods using control point networks

Table 6.1 presents a comparison of UPR routes which are optimised vertically only, horizontally only, and both vertically and horizontally. The routes are compared with each other and with the original flight plans, in two flight scenarios. The first flight scenario is UA6 with all flights preferring to minimize time. The second is UA0 with all flights preferring to minimize discomfort. The bad weather scenario that affects all the UPR flights is used. “Method” is the optimisation methods which are vertically, or horizontally or both vertically and horizontally optimised routes using control point networks. “NetM-H” is the horizontally optimised routes using control point networks. “NetM-HV” is the vertical optimisation with the horizontal routes given by horizontal optimisation using control point networks. “NetM-VH” is the horizontal optimisation with the optimal cruising altitudes given by vertical optimisation. “FS” is one of the two flight scenarios (“Pref Time” or “Pref DC”).

"Pref Time" and "Pref DC" correspond to flight scenarios UA6 and UA0. "Ave" is the reduction of time, fuel, or discomfort in average for a flight depending on which merge column ("Time", "Fuel", and "Discomfort") it belongs to. "Std" is the standard deviation of the reduction of time, fuel, or discomfort.

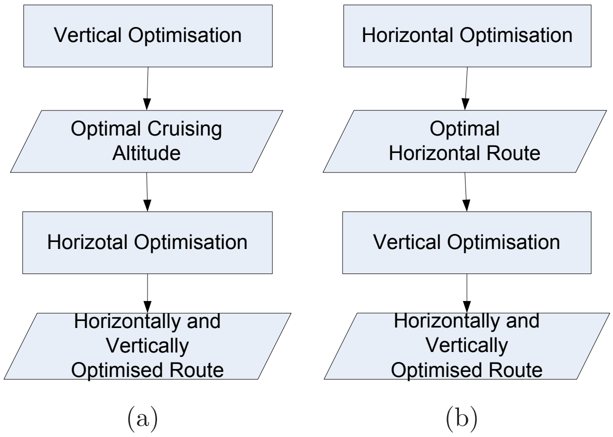


Figure 6.11: Horizontal and vertical optimisation.

The table shows that UPR routes optimised both horizontally and vertically are notably better than those only optimised vertically or horizontally. For example, for "Pref Time" flight scenario, on average "NetM-HV" and "NetM-VH" can reduce 13.11 and 12.97 minutes of time travelled, and 346.26 and 379.48 kg of fuel burn respectively, while "NetM-V" and "NetM-H" can only reduce 10 and 3.47 minutes of time travelled and 216.65 and 147.87 kg of fuel burn respectively. For this "Pref Time" flight scenario, on average "NetM-HV" and "NetM-VH" can also reduce the discomfort of 9.56 and 9.26, while "NetM-V" can reduce only 9.26 and "NetM-H" increases discomfort by 3.73. For the flight scenario "Pref DC", "NetM-HV" and "NetM-VH" can averagely reduce discomfort by 32.10 and 31.98, while "NetM-V" and "NetM-H" can reduce discomfort by 32.10 and 31.11. For this "Pref DC" scenario, "NetM-HV" and "NetM-VH" can also averagely reduce the time travelled by 9.56 and 2.63 and reduce fuel burn by 271.77 and 126.48 respectively, while "NetM-V" increases time travelled by 1.71 minutes and fuel burn by 74.06 kg and "NetM-H" can reduce time travelled by 1.66 minutes and fuel burn by 86.39 kg only. Here the decrease of discomfort can be roughly mapped to the time period that a flight passing through weather cells can save as one third of the discomfort value

Method	FS	Time Ave	Time Std	Fuel Ave	Fuel Std	Discomfort Ave	Discomfort Std
NetM-V	Pref Time	-10.00	15.50	-216.65	1062.90	-9.26	28.60
	Pref DC	1.71	24.50	74.06	1417.22	-32.10	28.22
NetM-H	Pref Time	-3.47	3.45	-147.87	208.04	3.73	14.87
	Pref DC	-1.66	4.42	-86.39	195.14	-31.11	27.89
NetM-HV	Pref Time	-13.11	16.12	-346.26	1118.17	-9.56	27.80
	Pref DC	-9.56	18.39	-271.77	1105.97	-32.10	28.22
NetM-VH	Pref Time	-12.97	15.97	-379.48	1133.39	-9.26	28.13
	Pref DC	-2.63	23.91	-126.48	1424.64	-31.98	28.05

Table 6.1: Comparison of different optimisation methods using control point networks.

when we assume that the bad weather level in all weather cells is 3 (the average bad weather level). For example if the discomfort reduction is 30 the time period the flight passing the bad weather areas can decrease by about 10 minutes. In terms of fuel reduction in percentage, when UPR flights prefer to minimize time, “NetM-V” and “NetM-H” can reduce by about 3% and 2% fuel burn in comparison with the original flight plans respectively; “NetM-HV” and “NetM-VH” can reduce about 5% fuel burn. These percentages are calculated based on the average fuel reductions per flight by these methods and the total fuel burn of 986958 kg by original flight plans (which will be presented in Table 6.5).

We also use T-Test to assess the statistical significance of differences between every two combinations of a method and a flight scenario. Tables 6.2, 6.3, and 6.4 show the T-Test values for time, fuel, and discomfort objective respectively. “Method-FS” is the combination of a method which is “NetM-V”, “NetM-H”, “NetM-HV”, or “NetM-VH” with a flight scenario which is “Pref Time” or “Pref DC”. For example, “V-Time” is the combination of “NetM-V” method and “Pref Time” flight scenario.

Assume that the difference assessment is for variable (objective) X. X_1 is the variable of the first method and X_2 is the variable of the second one. The T-Test value is calculated by Equation 6.2.

$$\frac{signal}{noise} = \frac{\overline{X_1} - \overline{X_2}}{SE(\overline{X_1} - \overline{X_2})} \tag{6.2}$$

The top part of the formula is the difference between the means of variables of the two methods. To compute it, we take the variance for each method and divide it by

Method-FS	V-Time	V-DC	H-Time	H-DC	HV-Time	HV-DC	VH-Time	VH-DC
V-Time	0.00	4.82	4.91	6.17	-1.65	0.22	-1.59	3.08
V-DC	-4.82	0.00	-2.49	-1.61	-6.02	-4.38	-5.98	-1.51
H-Time	-4.91	2.49	0.00	3.84	-6.97	-3.88	-6.93	0.41
H-DC	-6.17	1.61	-3.84	0.00	-8.16	-4.98	-8.14	-0.48
HV-Time	1.65	6.02	6.97	8.16	0.00	1.73	0.07	4.33
HV-DC	-0.22	4.38	3.88	4.98	-1.73	0.00	-1.67	2.74
VH-Time	1.59	5.98	6.93	8.14	-0.07	1.67	0.00	4.29
VH-DC	-3.08	1.51	-0.41	0.48	-4.33	-2.74	-4.29	0.00

Table 6.2: T-Test values for time objective.

the number of flights. We add these two values and then take their square root as in Equation 6.3, where the variance is the square of the standard deviation. Here n_1 is the same to n_2.

$$SE(\overline{X_1} - \overline{X_2}) = \sqrt{\frac{var_1}{n_1} + \frac{var_2}{n_2}} \tag{6.3}$$

we find a tabulated t value of 1.96 in t table with 282 degree of freedom (141 for n_1 + 141 for n_2) and probability of 0.05. In Tables 6.2, 6.3, and 6.4, $v[i, j]$ (the T-Test value of cell at row i and column j) is $-v[j, i]$, as $v[i, j]$ is the T-Test value between method i and method j, while $v[j, i]$ is that between method j and method i. However when the assessment of the significance of differences is based on the absolute values of these values.

The results show that most of the significant differences happen when one method minimizes time and the other minimizes discomfort and they becomes more significant when one optimises only horizontal or vertical dimensions and the other optimises both. As shown in these tables, the two methods with the most significant difference in time are "HV-Time" and "H-DC", followed by "VH-Time" and "H-DC", and "VH-Time" and "V-DC". The two couples of methods with the most difference in fuel are "VH-Time" and "H-DC", and "VH-Time" and "V-DC". Those for discomfort are "VH-DC" and "H-Time", "HV-DC" and "H-Time".

It can also be seen that the difference in discomfort is highest, followed by time and fuel with the T-Test maximum values of 13.40, 8.16, and 3.04 respectively.

Methods	V-Time	V-DC	H-Time	H-DC	HV-Time	HV-DC	VH-Time	VH-DC
V-Time	0.00	1.96	0.76	1.44	-1.00	-0.43	-1.25	0.60
V-DC	-1.96	0.00	-1.85	-1.34	-2.77	-2.29	-2.98	-1.19
H-Time	-0.76	1.85	0.00	2.57	-2.08	-1.31	-2.40	0.18
H-DC	-1.44	1.34	-2.57	0.00	-2.73	-1.97	-3.04	-0.33
HV-Time	1.00	2.77	2.08	2.73	0.00	0.56	-0.25	1.45
HV-DC	0.43	2.29	1.31	1.97	-0.56	0.00	-0.81	0.96
VH-Time	1.25	2.98	2.40	3.04	0.25	0.81	0.00	1.66
VH-DC	-0.60	1.19	-0.18	0.33	-1.45	-0.96	-1.66	0.00

Table 6.3: T-Test values for fuel objective.

Methods	V-Time	V-DC	H-Time	H-DC	HV-Time	HV-DC	VH-Time	VH-DC
V-Time	0.00	-6.77	4.80	-6.52	-0.09	-6.77	0.00	-6.76
V-DC	6.77	0.00	13.39	0.30	6.78	0.00	6.83	0.04
H-Time	-4.80	-13.39	0.00	-13.14	-5.02	-13.39	-4.87	-13.40
H-DC	6.52	-0.30	13.14	0.00	6.52	-0.30	6.57	-0.26
HV-Time	0.09	-6.78	5.02	-6.52	0.00	-6.78	0.09	-6.77
HV-DC	6.77	0.00	13.39	0.30	6.78	0.00	6.83	0.04
VH-Time	0.00	-6.83	4.87	-6.57	-0.09	-6.83	0.00	-6.81
VH-DC	6.76	-0.04	13.40	0.26	6.77	-0.04	6.81	0.00

Table 6.4: T-Test values for discomfort objective.

6.4.3 Summary

In this section, we present methods to optimise aircraft route both vertically and horizontally. The results show that UPR routes provided by these methods are noticeably better than those only optimised vertically or horizontally. Besides that by using T-Test to assess the significance of differences we found that the difference between a method minimize time and other minimizes discomfort is significant, and it is more significant if one optimises only vertically or horizontally and the other optimises both.

6.5 Comparison of NetM, GA, and LCS methods

In this section, we present the running time, the quality of UPR routes, and conflict detection resolution when UPR routes are found by NetM. A comparative reference to the results provided by GA and LCS is also presented. Then a systematic use of UPR methods is proposed.

6.5.1 Running Time

The experiments with NetM are step up with the same computing capability, the weather scenario, the flight scenario with GA and LCS as presented in Section 5.5.1. NetM is also set up to run in parallel similarly to GA method. When a computation node is free, NetM will be used to find a 3-D UPR-route for a flight (that hasn't yet got a UPR route).

The experiments show that the methods using network are fastest. They take only about 8 minutes by using 2 computation nodes to generate 142 3-D UPR routes (6.07 minutes and 8.96 minutes for GRP and GAP respectively); the running time is spent mainly on generating networks, the time to find the UPR route in a network is just a second. If the methods run in 8 computation nodes, NetM will take only about 2 minutes. These running times are obviously acceptable for UPR strategic routing (UPR route from origin to destination).

Further, NetM is especially suitable in tactical routing (which is implemented in real time) or in decentralized routing environment (which is implemented for the aircraft itself). When NetM is installed in an aircraft and the flight crew can use it to choose the preferred route for the aircraft itself, NetM will take only about 10 seconds to find a UPR route for a flight.

The running time for conflict detection and resolution between UPR flights (after having found UPR routes by NetM) with themselves and with non-UPR flights is the same to those routes provided by GA and LCS (about 30 seconds).

6.5.2 Quality of UPR routes

Table 6.5 presents a comparison of UPR methods using control point networks with each other and with the original flight plans in two flight scenarios. The first flight scenario is UA6 with all flights preferring to minimize time. The second is UA0 with all flights preferring to minimize discomfort. The bad weather scenario that affects all the UPR flights is used. "M & P" is the combinations of the method

		Time		Fuel		Discomfort	
M & P	FS	Value	%Diff	Value	%Diff	Value	%Diff
F Plan	F Plan	20116.9	0.00%	986958.1	0.00%	4558.7	0.00%
NetM-GRP	Pref Time	18696.3	-7.06%	956193.9	-3.12%	3243.4	-28.85%
	Pref DC	20360.0	1.21%	997474.3	1.07%	0.0	-100.00%
NetM-GAP	Pref Time	19624.7	-2.45%	965960.4	-2.13%	5088.9	11.63%
	Pref DC	19881.6	-1.17%	974690.5	-1.24%	141.3	-96.90%

Table 6.5: Time, Fuel, and Discomfort of the original flight plans and UPR routes in different flight scenarios.

(NetM) and a problem which is GRP, or GAP. “FS” is flight scenarios. “NetM - GRP” and “NetM-GAP” are the combinations of the method NetM with the problem GRP and GAP respectively. “F Plan” is the original flight plans. “Pref Time” and “Pref DC” correspond to flight scenarios UA6 and UA0. “Value” is the values of time, fuel, or discomfort objective depending on which merge column (“Time”, “Fuel”, and “Discomfort”) it belongs to. “%Diff” is the difference in percentage between the objectives provided by UPR methods and the ones provided by the original flight plans.

With reference to Table 5.15 it can be seen that NetM and GA can provide solutions with higher quality than LCS. NetM and GA can provide solutions that are almost similar, except for the case the flights are given routes (problem GRP) and the flight scenario is UA6. In this case, the discomfort objective of solutions provided by GA is almost equal to the one of solutions provided by NetM (14.9 against 0) but the time objective of solutions provided by GA is notably smaller than the one of solutions provided by NetM (18845.2 against 20360.0).

Table 6.5 shows that when the flight scenario is “Pref DC”, NetM can reduce the discomfort objective can reduce by more than 95% in comparison with the original flight plans; when the flight scenario is “Pref Time”, NetM can reduce the time objective by more than 2% in every problem-method combination and the fuel burn can be reduced by about 2%.

6.5.3 Conflict Detection and Resolution

After having found UPR routes by NetM, KB3D is applied to detect conflicts between these routes with each other and with Non-UPR routes. Conflicts are then resolved by departure time adjustment. The departure deviation results of UPR routes provided by NetM are similar to those provided GA and LCS. The total deviation of departure time is about 65 minutes; and the maximum departure time deviation of a flight is 2 minutes.

6.5.4 Proposal of a systematic use of UPR methods

NetM is the best to find UPR routes because of its short running time and the quality of solutions, especially in the context where UPR routes need to be updated with new traffic situations.

As discussed above GA in general can provide solutions as good as NetM in terms of the quality of UPR routes, but because of its long running time disadvantage, GA is suitable to find strategic UPR routes. Nevertheless GA can be installed in each aircraft to find the UPR route for only the aircraft itself in a decentralized routing environment. The time consumption in this case is also acceptable. GA will take about 13 seconds to find the UPR route by using 8 computation nodes.

LCS can be used to find a set of rules from the first stage or when the traffic environment updates. Air traffic controllers can take this set of rules into account while they are handling the conflicts so that they not only resolve the conflicts but also advise maneuvers following the set of rules to optimise aircraft routes.

The best classifier can be also found for a single flight, when LCS runs with the set of only one UPR flight. However, this best classifier can be used by only air traffic controllers, dispatcher and pilots responsible for that flight. This classifier can provide the UPR route of the flight with better quality than one that provides navigation for all UPR flights.

In summary, all methods (GA, LCS, NetM) can be used to find strategic UPR routes for flights before departure, as strategic planning can be implemented before departure a number of hours. Since tactical planning (adjusting UPR routes in real time) requests fast time computation, NetM is the most suitable for finding tactical UPR routes. Another option for GA to be applied in tactical planning is that GA can be installed in the cockpit of an aircraft to find the UPR route in real time for the aircraft (itself only). The set of rules provided by LCS can be taken into account while air traffic controllers decide maneuvers to resolve the conflicts among aircraft.

6.5.5 Conclusion

In conclusion, this section provides a comparison of 3 methods for finding UPR routes in terms to running time, and quality of solutions, and propose a reasonable application of the 3 methods in air traffic management operation. NetM are approximately twenty times faster than GA and LCS. While NetM, and GA provide better solutions than LCS, the set of rules provided by LCS can be reused for other sets of UPR flights, and can be used in deciding maneuvers for conflict resolution among flights. The departure time adjustments of UPR route sets provided by these methods are small, and almost the same (about 65 minutes for the sets of 142 UPR routes).

6.6 Emission Analysis of UPR routes using NetM

In this section we present and compare emission distributions of UPR routes provided by the different methods using control point networks. The emission distributions by methods “NetM-HV” and “NetM-VH” described in Section 6.4 are presented and compared first, followed by those by methods “NetM-V” and “NetM-H”. The emission distribution analysis is used models which are presented in Section 4.7.

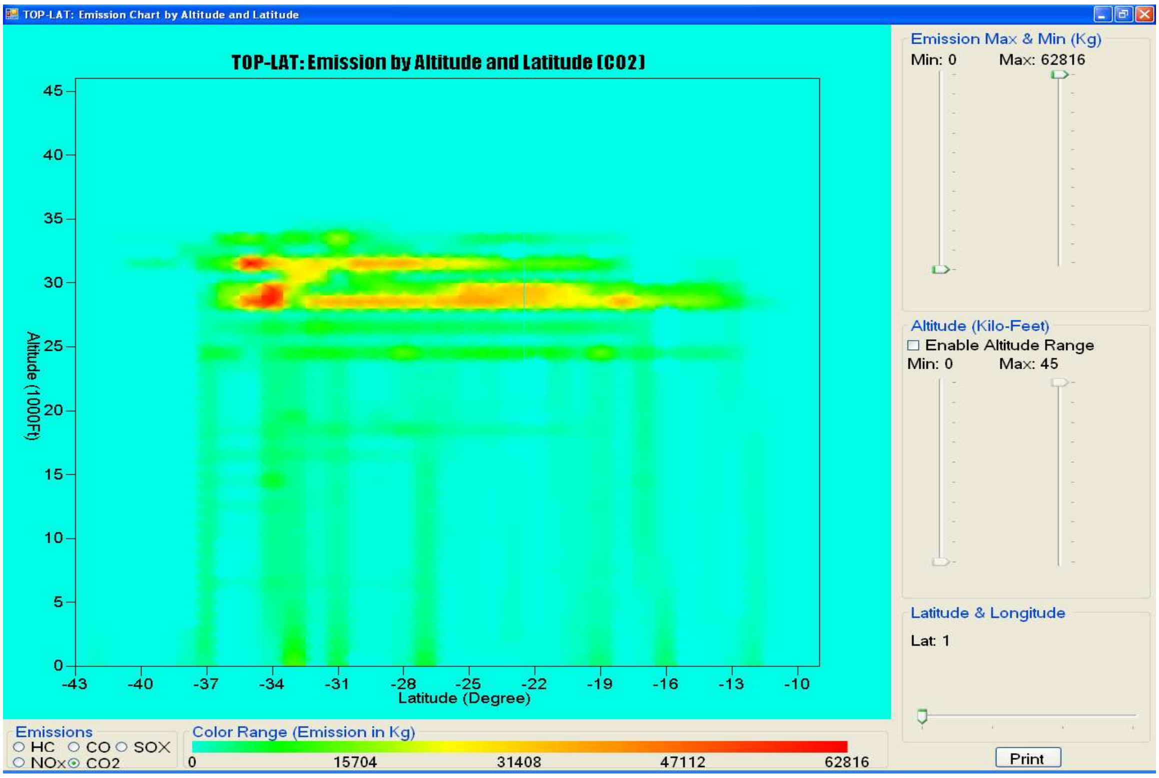
Figures 6.12, 6.13, 6.14, and 6.15 show CO_2 and CO emission distributions by altitude and latitude for the same set of UPR flights, when UPR routes are found by methods “NetM-HV” and “NetM-VH”. Figures 6.12 and 6.14 show the emission distributions when UPR flights prefer to minimize time. Figures 6.13 and 6.15 present the emission distributions when UPR flights prefer to minimize discomfort. Each figure has 2 sub-figures. The upper sub-figure presents the emission distributions when UPR flights optimise their routes vertically based on the horizontal routes that are provided by horizontal optimisation (this method is "NetM-HV"). The lower one presents the emission distributions when UPR flights optimise their routes horizontally using the optimal cruising altitudes that are provided by vertical optimisation (this method is "NetM-VH").

Figures 6.12 and 6.14 show that when UPR flights prefer to minimize time, the emission distributions by "NetM-HV" and "NetM-VH" are similar. Particularly, CO_2 is mainly from 25,000 feet to 33,000 feet; CO_2 in lower altitudes is not significant (presented in Figure 6.12). For CO emission, in addition to the density in the altitude range from 25,000 feet to 33,000 feet there is quite significant distribution of CO in the lower altitude (presented in Figure 6.14), though the time the aircraft is at the lower altitude range is not much. This is because aircraft produce significant amounts of CO at low altitude.

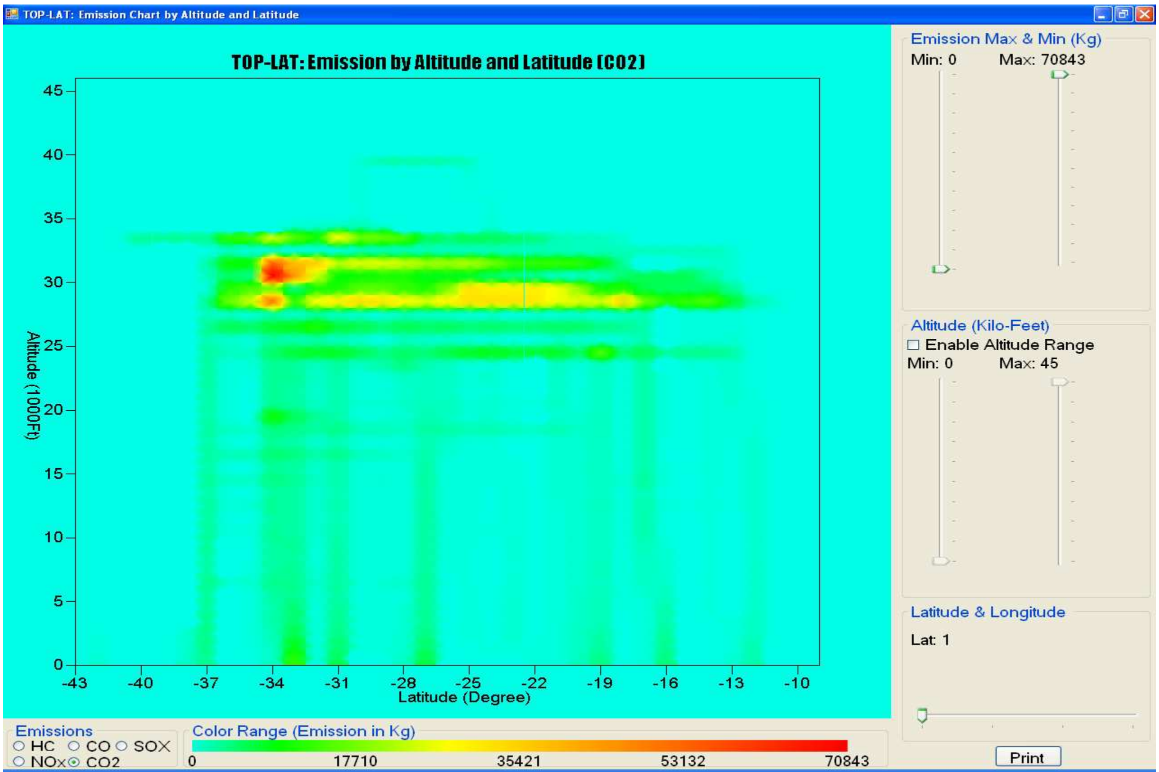
However, Figures 6.13 and 6.15 show that when UPR flights prefer to minimize discomfort, the emission distributions by "NetM-HV" and "NetM-VH" are different. The emission distributions (for CO_2 in Figure 6.13 and particularly for CO in Figure 6.15) by "NetM-HV" are mainly from 25,000 feet to 33,000 feet, while the emission distributions by "NetM-VH" are from this altitude range and also from 10,000 to 15,000 feet. The reason is UPR flights choose various altitudes to avoid bad weather cells when discomfort minimization is more preferred.

The horizontal emission distributions of UPR routes when UPR flights prefer to minimize time or discomfort for both "NetM-HV" and "NetM-VH" are all similar. The high density of emissions is around main airports and the coast line. Figure 6.16 is one example of horizontal distribution of CO_2 when UPR flights prefer to minimize time.

For user preferred routes that are provided by "NetM-V" (vertical optimisa-

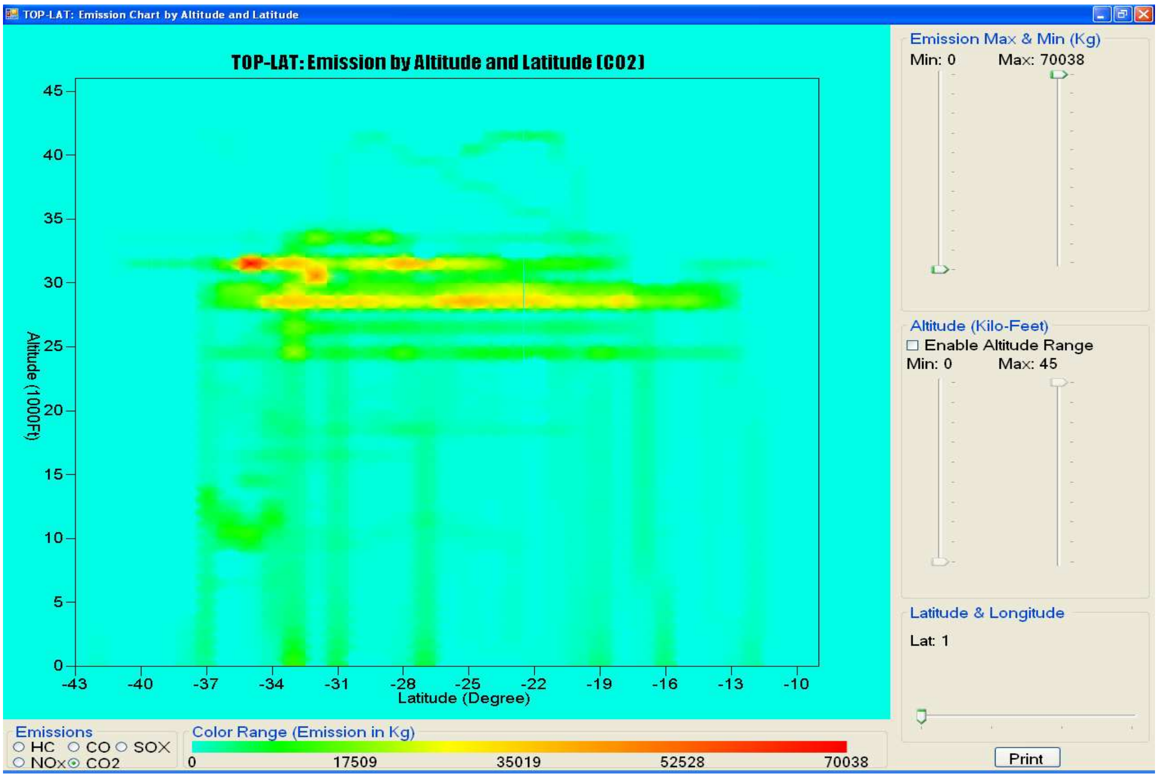


(a) CO_2 by altitude and latitude aggregated longitudinally, NetM-HV.

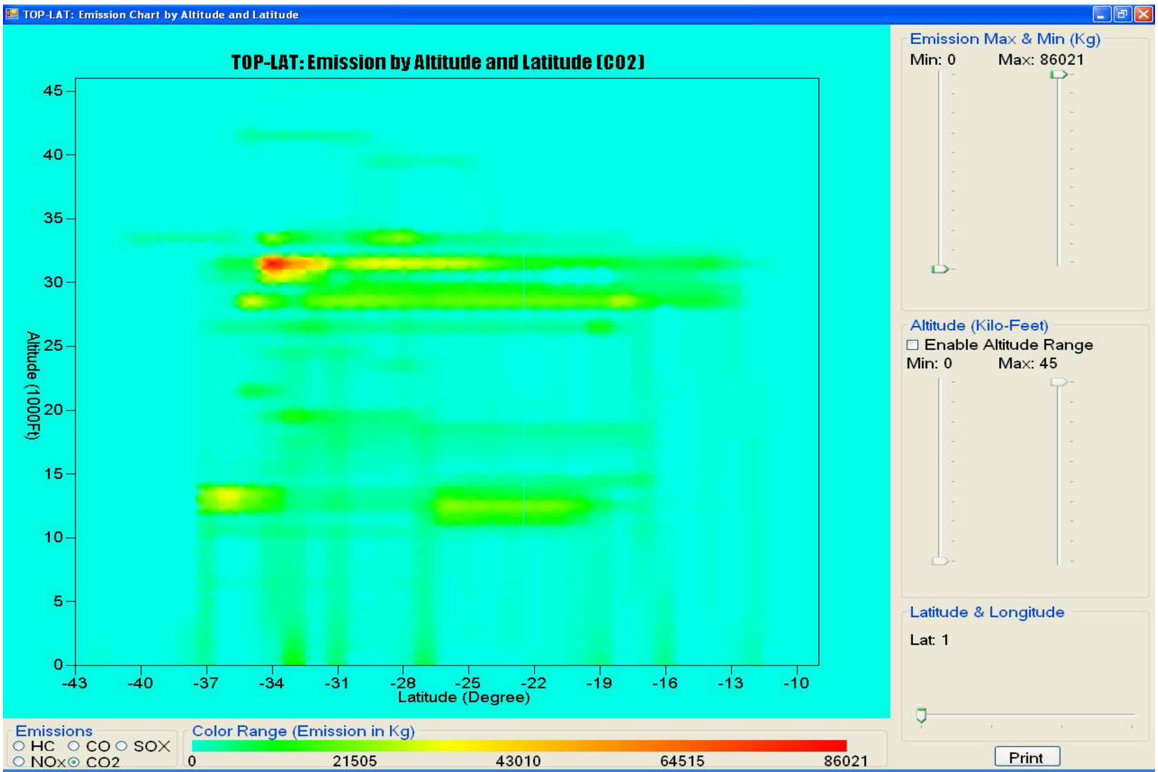


(b) CO_2 by altitude and latitude aggregated longitudinally, NetM-VH.

Figure 6.12: CO_2 by altitude and latitude aggregated longitudinally, minimal time routes.

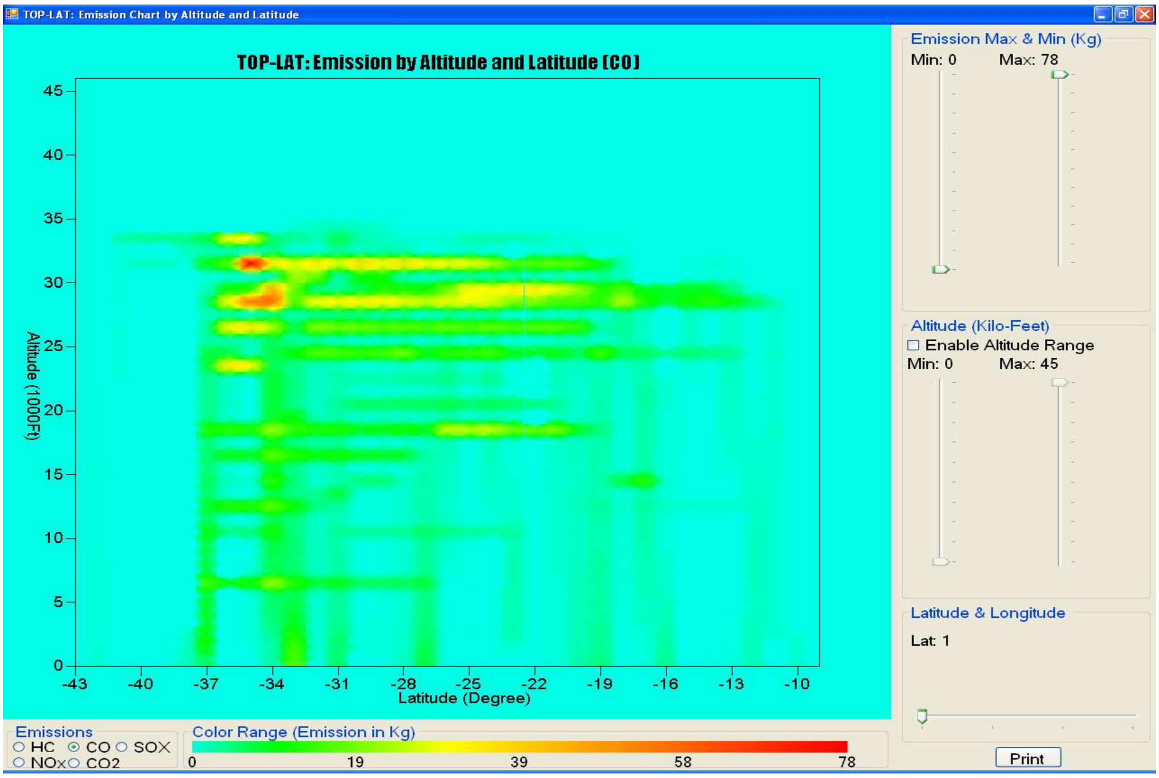


(a) CO_2 by altitude and latitude aggregated longitudinally, NetM-HV.

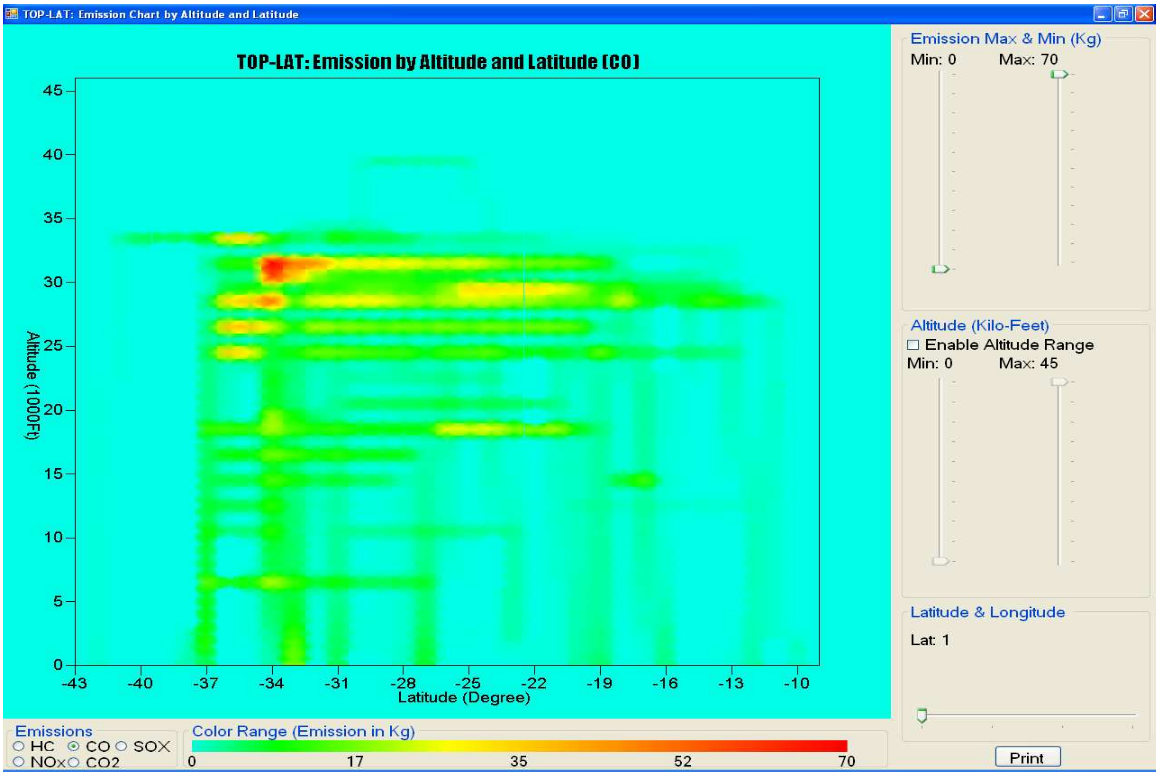


(b) CO_2 by altitude and latitude aggregated longitudinally, NetM-VH.

Figure 6.13: CO_2 by altitude and latitude aggregated longitudinally, minimal discomfort routes.

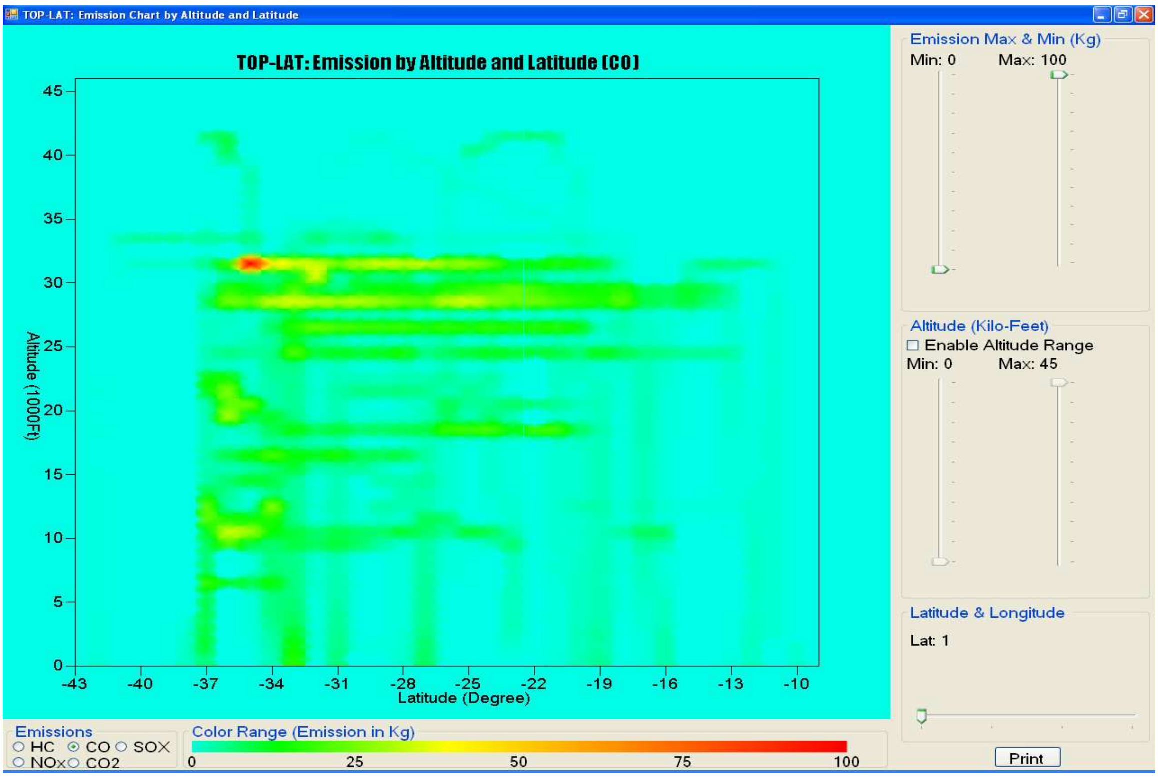


(a) CO by altitude and latitude aggregated longitudinally, NetM-HV.

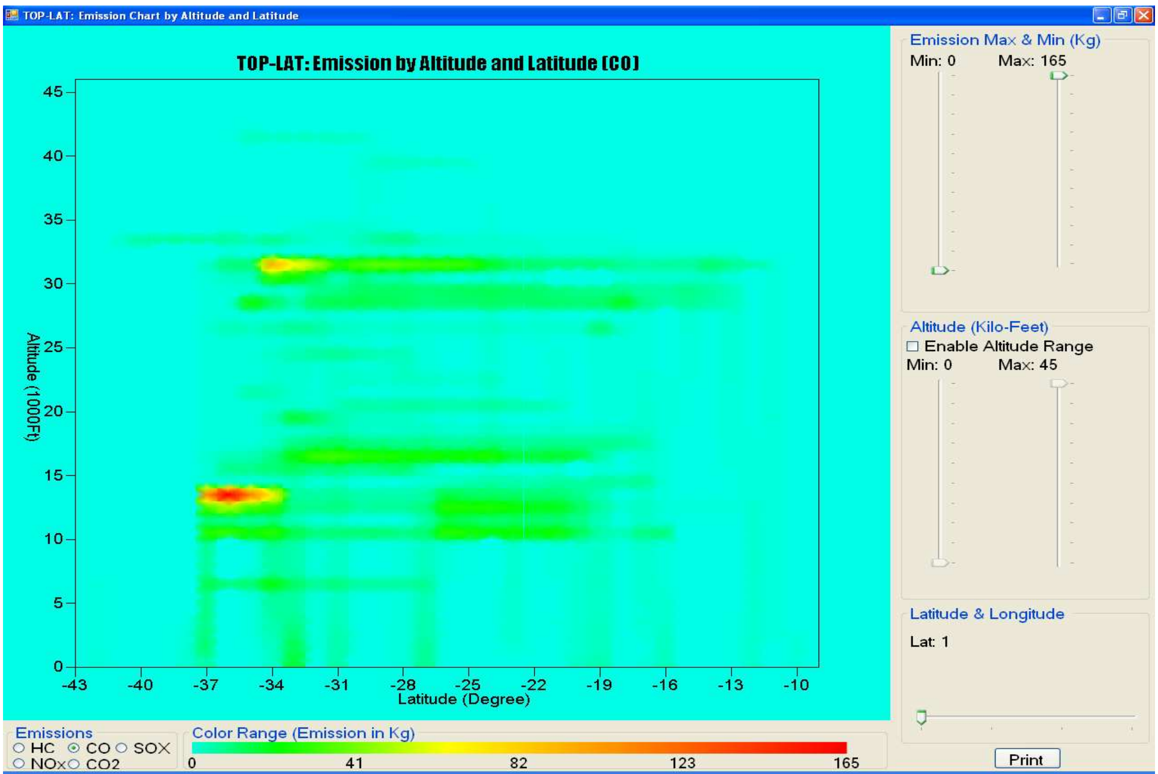


(b) CO by altitude and latitude aggregated longitudinally, NetM-VH.

Figure 6.14: CO by altitude and latitude aggregated longitudinally, minimal time routes.

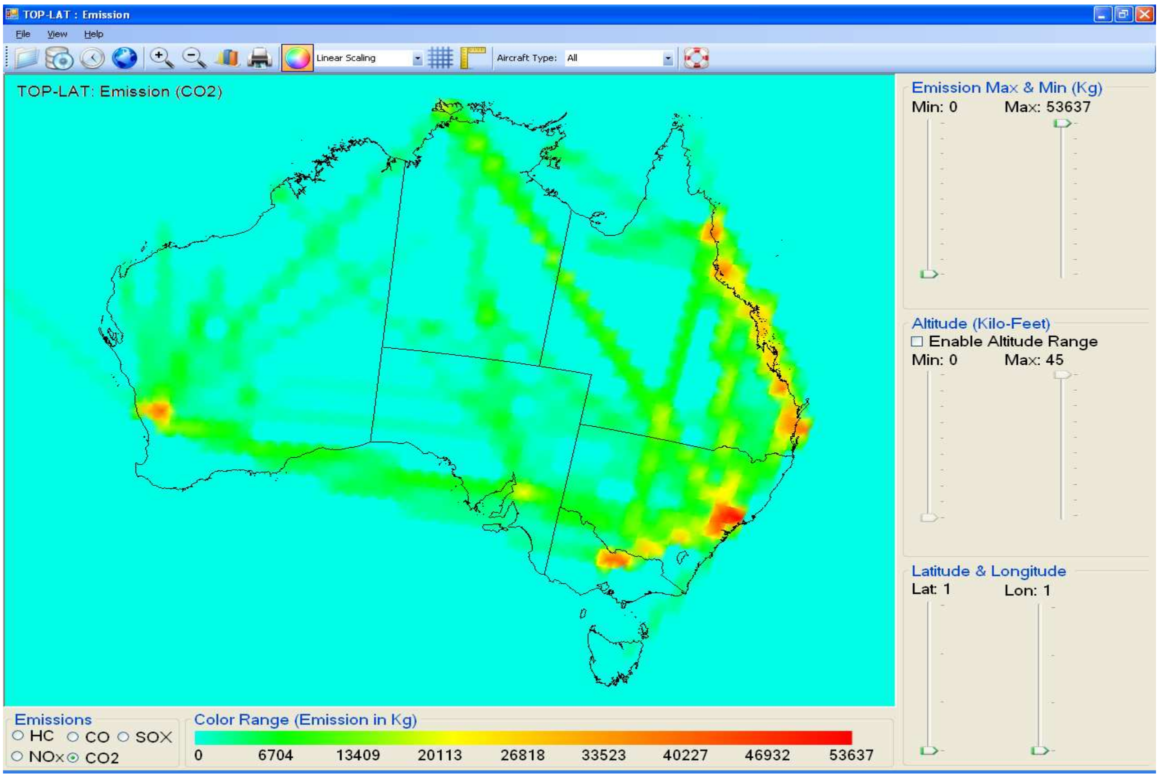


(a) CO by altitude and latitude aggregated longitudinally, NetM-HV.

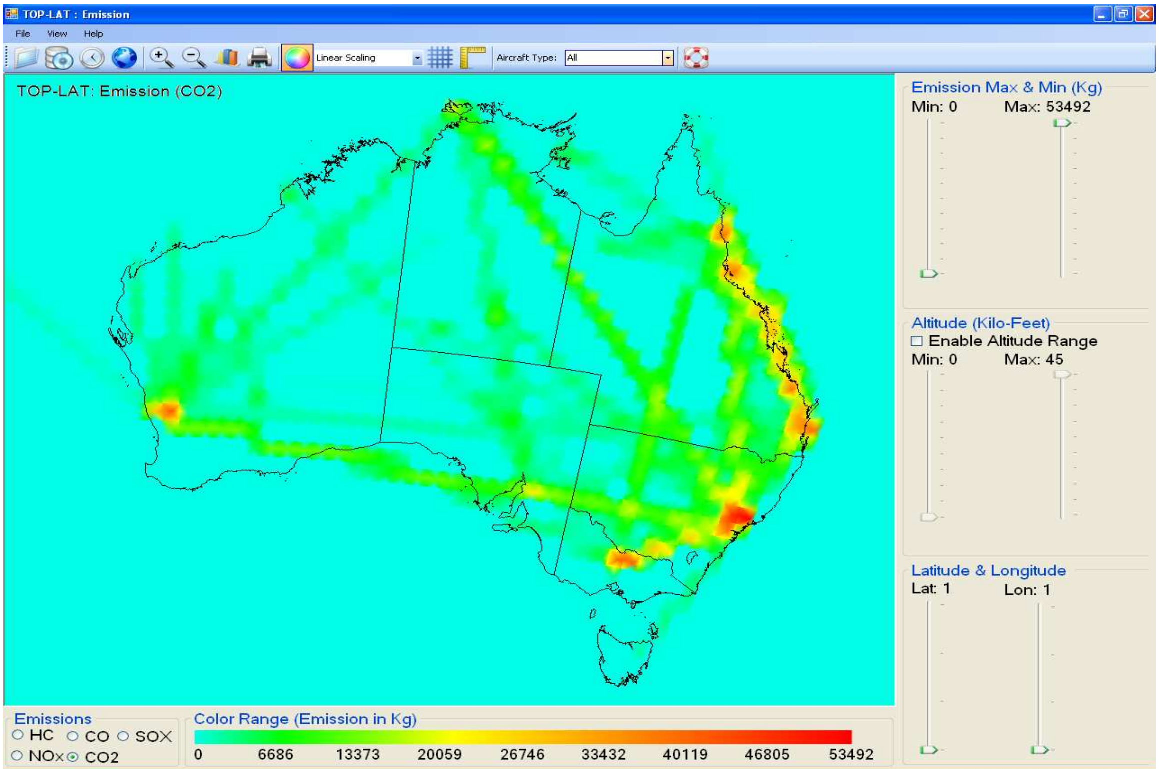


(b) CO by altitude and latitude aggregated longitudinally, NetM-VH.

Figure 6.15: CO by altitude and latitude aggregated longitudinally, minimal discomfort routes.

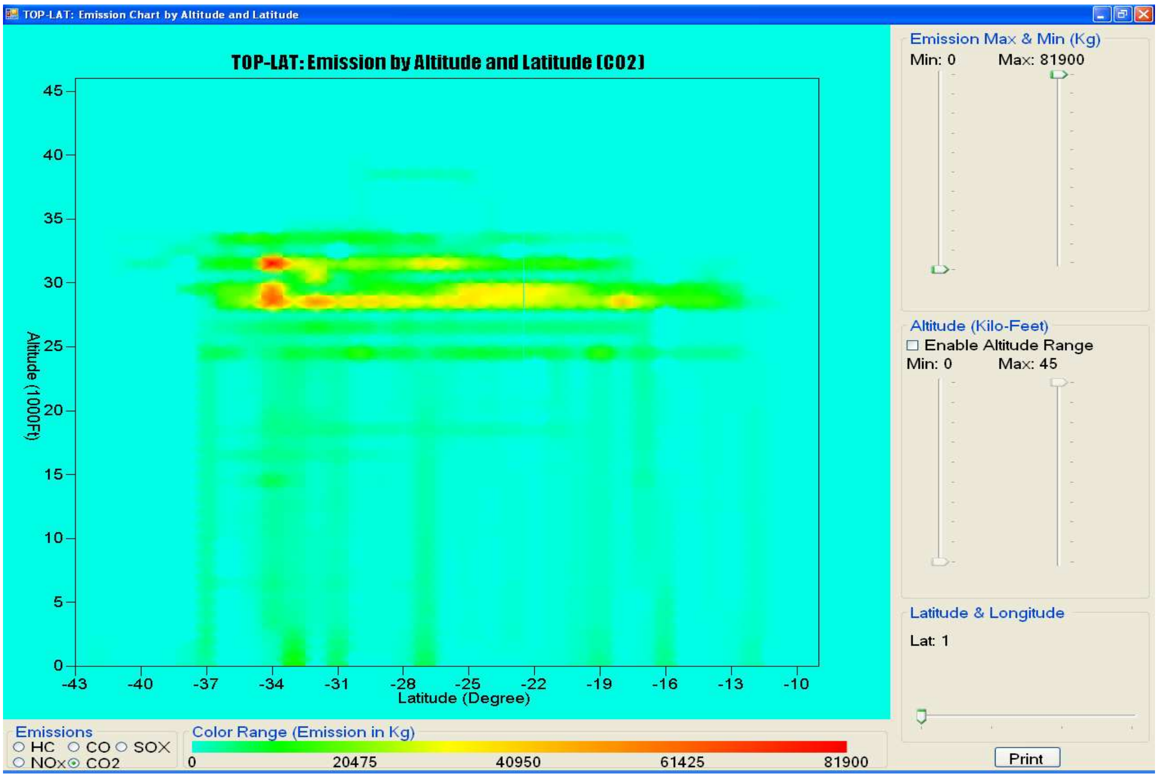


(a) CO_2 by longitude and latitude aggregated from ground level to 45 kf, NetM-HV.

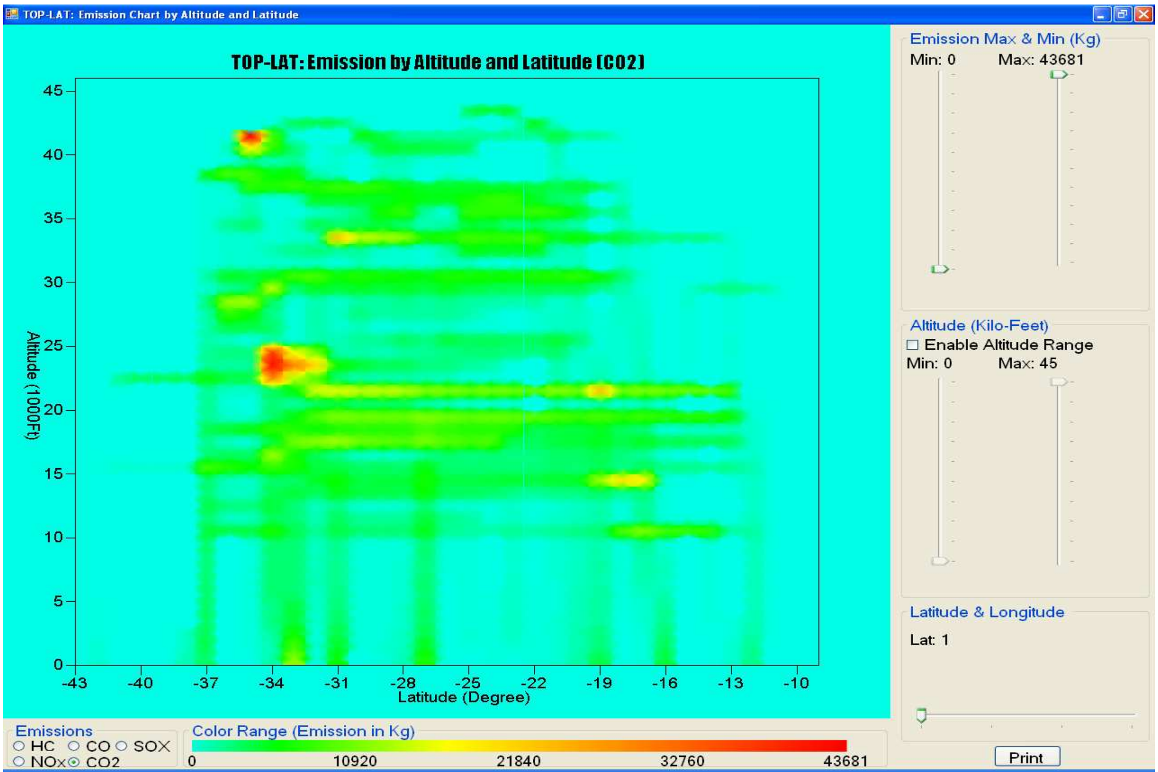


(b) CO_2 by longitude and latitude aggregated from ground level to 45 kf, NetM-VH.

Figure 6.16: CO_2 by longitude and latitude aggregated from ground level to 45 kf, minimal time routes.

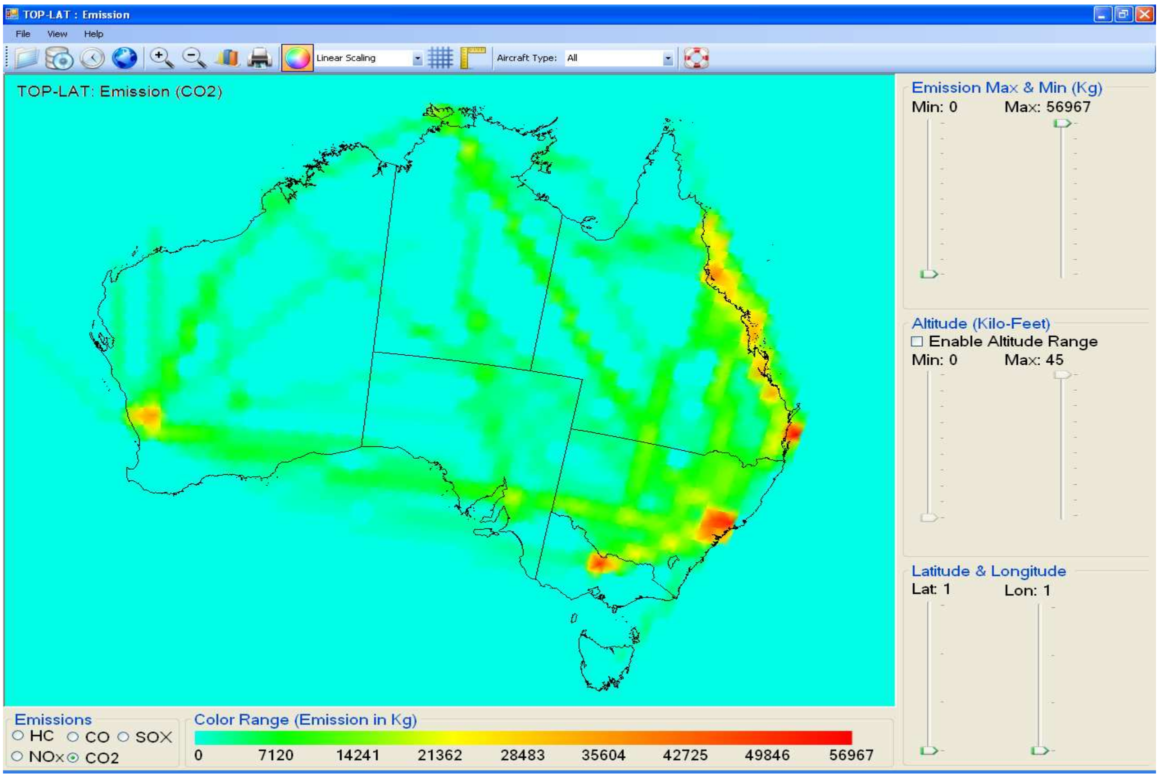


(a) CO_2 by altitude and latitude aggregated longitudinally, NetM-V.

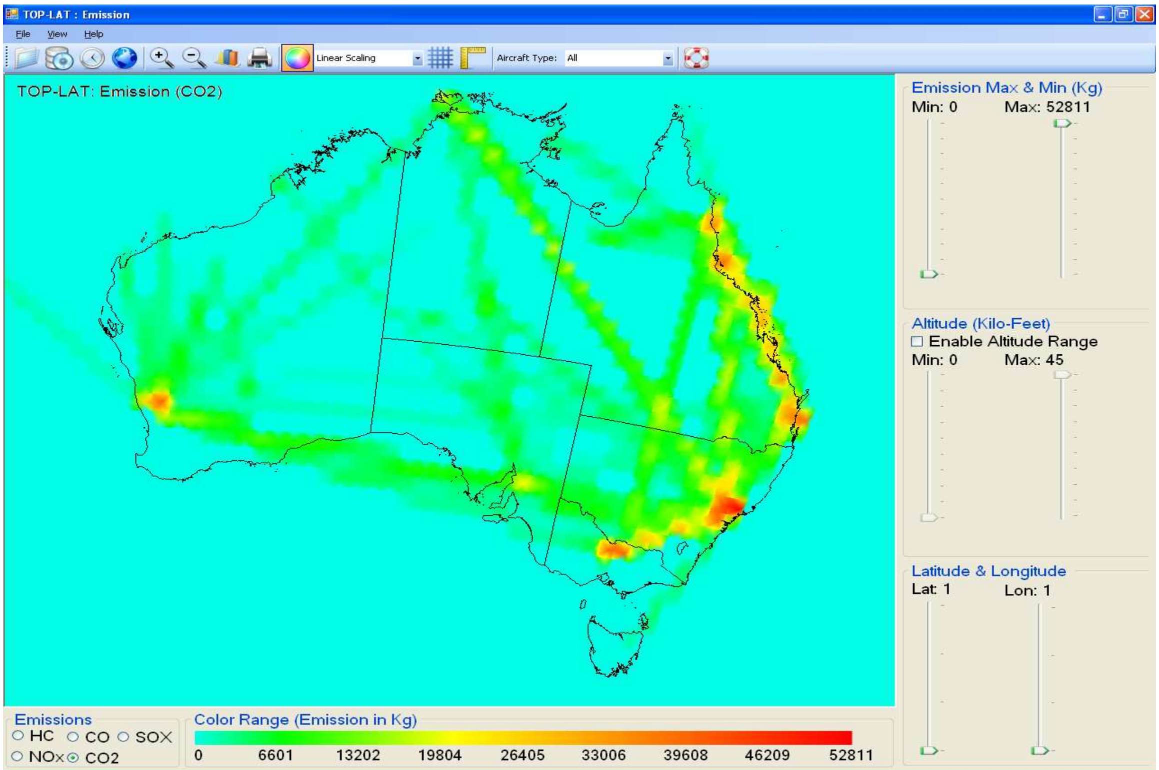


(b) CO_2 by altitude and latitude aggregated longitudinally, NetM-H.

Figure 6.17: CO_2 by altitude and latitude aggregated longitudinally, minimal time routes.



(a) CO_2 by longitude and latitude aggregated from ground level to 45 kf, NetM-V.



(b) CO_2 by longitude and latitude aggregated from ground level to 45 kf, NetM-H.

Figure 6.18: CO_2 by longitude and latitude aggregated from ground level to 45 kf, minimal time routes.

tion using original flight plans' routes), the horizontal emission distributions are even and the vertical emission distributions are as concentrated as those provided by "NetM-HV" and "NetM-VH". For user preferred routes which are provided by "NetM-H" (horizontal optimisation using original flight plans' cruising altitudes), the vertical emission distributions are even and the horizontal emission distributions are as concentrated as those provided "NetM-HV" and "NetM-VH" methods.

Figures 6.17 and 6.18 present the horizontal and vertical CO_2 distributions when UPR flights prefer to minimize time. The emission distributions of UPR routes provided by "NetM-V" and "NetM-H" methods are presented in the upper and lower sub-figures in each figure respectively. Figure 6.17 shows that CO_2 of UPR routes provided by "NetM-V" is vertically concentrated around 30,000 feet (in sub-figure 6.17(a)), while the one provided by "NetM - H" (in sub-figure 6.17(b)) is distributed evenly from 10,000 feet to 42,000 feet. Figure 6.18 shows the CO_2 emission of UPR routes provided by "NetM-V" (in sub-figure 6.18(a)) is horizontally more distributed than the one provided by "NetM-H" (in sub-figure 6.18(b)). The evidence is that there are more emission cells with high density in sub-figure 6.18(b) than in sub-figure 6.18(a).

6.7 Conclusion

In this chapter, we propose a network model for finding UPR routes and algorithms to generate networks. The proposed algorithms to generate control point networks for flights work effectively to find UPR routes for flights. Dijkstra's algorithm (with a modification to improve the speed of the algorithm) is applied to find UPR routes in networks. NetM method can find UPR routes in a short period of time. In the experiments they can find UPR routes for 142 flights in about 6 minutes using two computational nodes or about 3 minutes if 4 nodes are used. The running time of NetM is much less than the ones of GA and LCS methods. The solution provided by NetM can also satisfy user preference. For example, NetM can find almost discomfort free UPR routes for flights preferring to minimize discomfort. The results also show that discomfort and time objectives are proportional reasonably to flight utility allocations.

Chapter 7

Conclusion

7.1 Summary of results

This thesis designs and develops a framework for planning User Preferred Routes and evaluating UPR concept. The main contribution of the thesis is the design and development of three methods for User Preferred Routing, which deal with a large number of flights, taking into account wind, bad weather and aircraft performance data. The environmental impact of User Preferred Route sets, which are optimised vertically, horizontally, or both for time and passenger discomfort minimization is analysed. A proper use of user preferred routing methods for strategic or tactical routing, and centralized or decentralized routing is proposed to take advantages of each method.

To enable the investigation and optimisation of UPR, a simulation and evaluation environment is established. In this environment, I developed a real time weather system to retrieve and process weather data. This data is then stored in a database for aviation decision support in general and for aircraft UPR in particular. Second, flight and weather scenarios are designed to test UPR methods and assess user preferred routes provided by these methods. Third, a segment based simulation environment is developed to simulate any type of route segment (climb, cruise, or descent). The simulation uses point mass model of aircraft, and is implemented in a continuous environment, as well as in a fast time mode. This simulation environment is applied effectively and flexibly to evaluate UPR.

Different emissions are measured effectively and their impact on the environment is analysed. I developed a real time flight data management system with algorithms to process, estimate, and integrate information for every flight in the airspace and to construct flight objects. Then another real time system is developed to estimate aviation emission using the flight objects provided by the flight data management system. The aviation emission is stored in 4-D database to analyse the impact of aviation emission on the environment. A number of models for emissions analysis are designed and implemented. The models developed here are then used for the calculation and analysis of fuel and emissions for UPR routes. From the spatial analysis of emissions data, we found that the CO_2 concentration in some parts of Australia is much higher than other parts, especially in some major cities. The emission results also show that NO_x emission of aviation may have a significant impact on the ozone layer in the upper troposphere, but not in the stratosphere. It is expected that with the availability of this real time aviation emission database, environmental analysts and aviation experts will have an indispensable source of information for making timely decisions regarding expansion of runways, building new airports, applying route charges based on environmentally congested airways, and restructuring air traffic flow to achieve sustainable air traffic growth.

This thesis proposes two approaches using evolutionary algorithms to resolve the UPR problems. One is the black box approach (which uses Genetic Algorithms) and the other is the white box approach (which uses Learning Classifier Systems). The mutation probability investigation shows that GA can work better than random search. The results show that the best solutions for flights preferring to minimize discomfort can be found faster than the ones for flights preferring to minimize time. The reason is the minimum discomfort route is usually found very fast. Then GA can focus on improving the time objective of solutions whose discomforts are minimum. Though LCS methods cannot provide solutions as good as "black box" methods using GA, its solutions are transparent to users, and reusable. LCS is also able to discover rules that can be applied to operation in air traffic control. These rules may be not yet known in air traffic management.

In this thesis, I designed and developed algorithms for UPR planning, based on dynamically generated networks of control points. This method is named as NetM. Two models of control point networks are proposed, for a flight which is given either a route or a cruising altitude. The algorithms to generate a network corresponding to a network model are designed. The segment simulation base is applied to determine the nodes and weights of links in a network. A network generated for a combination of origin, destination, aircraft model, route/cruising altitude, and weather data can be used for all the flights with the same combination. The algorithms work effectively to find UPR routes for flights. They can find UPR routes in a short period of time. The solution provided by these algorithms can also satisfy user preference. For example, they can find almost discomfort free UPR routes for flights preferring to minimize discomfort. The results also show that discomfort and time objectives are proportional reasonably to flight utility allocations.

In general, NetM (the methods using control point network) is the fastest compared to GA (the methods using Genetic Algorithms) and LCS (the methods using Learning Classifier Systems). If these methods run in 8 computation nodes, NetM will take only about 2 minutes and GA and LCS will take about 30 minutes for finding user preferred routes for 142 flights. These running times are acceptable in UPR strategic planning.

NetM and GA can provide solutions with higher quality than LCS. The UPR methods using networks of control points are also used to optimise aircraft route both vertically and horizontally. UPR routes optimised both horizontally and vertically are notably better than those only optimised vertically or horizontally. In terms of fuel reduction in percentage, “NetM-V” and “NetM-H” can reduce by about 3% and 2% fuel burn in comparison with the original flight plans respectively; “NetM-VH” and “NetM-HV” can reduce about 5% fuel burn.

Flight and weather scenarios are designed and generated to test and assess the performance of UPR methods. The advantages and disadvantages of UPR algorithms are pointed out. A systematic use of UPR methods is proposed for specific routing circumstances (strategic or tactical routing, and centralized or decentralized routing) to take the advantages of each method. In summary, all methods (GA, LCS, and NetM) can be used for strategic UPR routing for flights before departure. NetM is the most suitable for finding tactical UPR routes. Another option for GA to be applied in tactical planning is that GA can be installed in the cockpit of an aircraft to find the UPR route in real time for the aircraft in decentralized routing. The set of rules provided by LCS can be taken into account while air traffic controllers decide maneuvers to not only resolve the conflicts among aircraft, but also to optimise operations.

The environmental impact of user preferred route sets is analysed by using the tools developed from the aviation emission calculation and analysis system. It is found that when UPR flights prefer to minimize time, the emission distributions by "NetM-HV" method (where user preferred routes are optimised horizontally then vertically) and "NetM-VH" method (user preferred routes are optimised vertically then horizontally) are similar. However, when UPR flights prefer to minimize discomfort, the emission distributions by "NetM-HV" and "NetM-VH" are different. When UPR flights prefer to minimize time we also found the following. For user preferred routes that are provided by "NetM-V" (vertical optimisation using original flight plans' routes), the horizontal emission distributions are even and the vertical emission distributions are as concentrated as those provided by "NetM-HV" and "NetM-VH". For user preferred routes which are provided by "NetM-H" (horizontal optimisation using original flight plans' cruising altitudes), the vertical emission distributions are even and the horizontal emission distributions are as concentrated as those provided by "NetM-HV" and "NetM-VH" methods.

An efficient method to detect conflicts between flights is designed. The conflicts are then resolved by changing the departure time. This helps to reduce the time, fuel, emissions, and ATC workload as the result of minimizing the probability of applying maneuvers in the air to resolve conflicts. The CDR algorithms work effectively. The running time for both conflict detection and resolution of 142 UPR flights with each other and with 1,674 non-UPR flights is only about 30 seconds. The adjustment time is only about 65 minutes. In this thesis, we actually propose to determine the departure times for UPR flights (the departure times of UPR flights have not been published), so that UPR flights can minimize the risk of conflicts in the air. However in the case the departure times of UPR flights have been published, the result of 65 minute adjustment shows that if UPR planning is applied, the delay in departure times of UPR and non-UPR flights is insignificant (the average delay is less than 1 minute). This means UPR concept will likely not affect on delays at airports. Moreover UPR planning with accurate trajectory prediction can help to reduce delays at airports by determining accurate departure times for flights.

7.2 Future directions

There are several possibilities to extend and improve the research presented in this thesis.

- Aircraft route simulation: The route simulation can be improved by trajectory and route smoothing techniques as presented in Section 2.3.3.1. This will help to improve the accuracy of simulation.
- Aviation emission analysis: Dispersion of emissions by weather phenomena and chemical interaction happening in the atmosphere will be considered. This will help to provide more accurate assessment and prediction of the distribution and the impact of aviation emission on the environment. From the analysis we may find quantitative functions representing the impact of aviation emissions on the environment. Then these functions can be integrated into User Preferred Routing. Such study has been carried in [173] to deal with the trade-off between persistent contrails formation and additional fuel consumption, as the result of discovery that persistent contrails may have a three to four times greater effect on the climate than CO_2 emissions [194].

- Algorithms for UPR planning can be extended by taking not only time and discomfort as objectives but also departure time adjustments. For example, the sequence of flights for resolving conflicts by adjusting their departure times can be optimised to obtain minimum departure time deviation.

- Currently UPR algorithms are designed to find UPR routes from origins to destinations in static weather environment. However these algorithms can be easily extended to find UPR from any point in the routes of flights to their destinations. They can also take updated weather and air traffic situation into account.

- Learning Classifier System may be improved by integrating more interactions, not just between flights and weather environment as moment. For example, the interaction between flights that may conflict with each other may be added into a rule.

- In the future, we expect to use posterior, and interactive approaches in order to compare with the current approach (priori). A more efficient method might be found. In these approaches, multi-objective theory is used to find the non-dominated set as well as to determine the most suitable solutions, based on user preference during the search or after the search.

- Other or a new design of evolutionary operators might be investigated to improve the performance of GA and LCS. A hybrid of GA or LCS with NetM can be another choice for improvement. For example, GA or LCS can be used to search user preferred routes in the networks of control points to minimize time travelled, discomfort, and also delay time.

- In the future we will consider sector capacity and departure and arrival time slots in UPR planning. Sector capacity is an important factor in aircraft route planning [41].

- UPR algorithms can be extended to connect with the ground actions of aircraft such as taxi, pushback, and fuel.

- We are particulary interested in continuing the evaluation of UPR concept on the environment and other perspectives such as delay, safety, etc, as well as its relation with and impact on the whole air traffic management system.

Bibliography

[1] H. A. Abbass, R. Sarker, and C. Newton. PDE: A Pareto frontier differential evolution approach for multiobjective optimization problems. In *the Congress on Evolutionary Computation*, volume 2, pages 971 – 978, Seoul Korea, 2001.

[2] R. Akbaria and K. Ziaratia. A multilevel evolutionary algorithm for optimizing numerical functions. *International Journal of Industrial Engineering Computations*, 2:419 – 430, 2011.

[3] S. Alam, H. A. Abbass, and M. Barlow. ATOMS: Air traffic operations and management simulator. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):209–225, 2008.

[4] S. Alam, H.A. Abbass, and M. Barlow. Multi-objective ant colony optimization for weather avoidance in a free flight environment. Technical report, The Artificial Life and Adaptive Robotics Laboratory, University of New South Wales, Canberra, Australia, 2006.

[5] S. Alam, M. H. Nguyen, H. A. Abbass, C. Lokan, M. Ellejmi, and S. Kirby. A Dynamic Continuous Descent Approach Methodology for Low Noise and Emission. In *29th IEEE/AIAA Digital Avionics Systems Conference*, pages 1.E.5–1 – 1.E.5–18, 2010.

[6] J. N. Amin, J. D. Boskovic, and R. Mehra. A Fast and Efficient Approach to Path Planning for Unmanned Vehicles. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Aug 2006.

[7] H. V. Anders. *User preferred trajectories in commercial aircraft operation design and implementation.* PhD thesis, Department of Aeronautical and Vehicle Engineering, Division of Aerodynamics, SE-100 44 Stockholm, Sweden, 2007.

[8] E.P. Anderson, R.W. Beard, and T.W. McLain. Real-Time Dynamic Trajectory Smoothing for Unmanned Air Vehicles. *IEEE Transactions on Control Systems Technology*, 13(3):471 – 477, 2005.

[9] M. Anderson, J. Lopez, and J. Evers. A Comparison of Trajectory Determination Approaches for Small UAVs. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Aug 2006.

[10] G. Andreatta and L. Brunetta. Multi-Airport Ground Holding Problem: A Computational Evaluation of Exact Algorithms. Technical report, Department of Pure and Applied Mathematics, University of Padova, Padova, Italy, 1995.

[11] G. Andreatta, A. R. Odoni, and O. Richetta. *Computation and Information Processing in Air Traffic Control*, chapter Models for the Ground-Holding Problem, pages 125 – 168. Springer-Verlag, Berlin, 1993.

[12] I. Arasaratnam and S. Haykin. Cubature Kalman Filters. *IEEE Transactions on Automatic Control*, 54(6):1254 – 1269, 2009.

[13] ASTRA. Australian ATM Strategic Plan Volume 2. Technical report, ASTRA-Australian Strategic Air Traffic Management, 2003.

[14] ASTRA. Air Traffic Management: A strategic vision for Australia, Part B Operational Evolution Document. Technical report, The Australian Strategic Air Traffic Management Group, 2007.

[15] J. Bacardit. *Pittsburgh Genetics-Based Machine Learning in the Data Mining era: Representations, generalization, and run-time.* PhD thesis, Ramon Llull University, Barcelona, Spain, 2004.

[16] R.H. Bartels and J.C. Beatty. *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling.* Morgan Kaufmann, San Francisco, CA, 1st edition, 1998.

[17] D. M. Bates and D. G. Watts. *Nonlinear Regression and Its Applications.* Wiley, New York, 1st edition, 1988.

[18] S. L. Baughcum, T. G. Tritz, S. C. Henderson, and D. C. Pickett. Scheduled Civil Aircraft Emission Inventories for 1992: Database Development and Analysis, NASA Contractor Report 4700. Technical report, NASA, Washington, D.C., 1996.

[19] J. Bellingham, A. Richards, and J.P. How. Receding horizon control of autonomous aerial vehicles. In *American Control Conference*, volume 5, pages 3741–3746. IEEE, 2002.

[20] J. Bellingham, M. Tillerson, A. Richards, and J. P. How. Multi-Task Allocation and Path Planning for Cooperating UAVs. In *Second Annual Conference on Cooperative Control and Optimization*, 2011.

[21] D. P. Bertsekas. *Nonlinear Programming* . Athena Scientific, Belmont, Massachusetts, 2nd edition, 1999.

[22] D. Bertsimas and A. Odoni. A Critical Survey of Optimization Models for Tactical and Strategic Aspects of Air Traffic Flow Management. Technical Report no. 19980004146, NASA, 1997.

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

[24] K. D. Bilimoria, B. Sridhar, G. Chatterji, K. S. Sheth, and S. Grabbe. FACET: Future ATM concepts evaluation tool. In *Proc. of 3rd USA/Europe Air Traffic Management Research and Development Seminar*, Napoli, Italy, 2000.

[25] C. M. Bishop and C. M. Roach. Fast curve fitting using neural networks. *Review of Scientific Instruments*, 63(10):4450 – 4456, 1992.

[26] H. Blom, G. Bakker, P. Blanker, J. Daams, M. Everdij, and M. Klompstra. *Air Transportation Systems Engineering*, chapter Accident risk assessment for advanced ATM, pages 463–480. AIAA Press, 2001.

[27] S. A. Bortoff. Path Planning for UAVs. In *Proceedings of the American Control Conference*, Jun 2000.

[28] G. P. Brasseur, R. A. Cox, D. Hauglustaine, I. Isaksen, J. Lelieveld, D. H. Lister, R. Sausen, U. Schumann, A. Wahner, and P. Wiesen. European scientific assessment of the atmospheric effects of aircraft emissions. *Atmospheric Environment*.

[29] D.J. Brudnicki and D.B. Kirk. Trajectory modeling for automated en route air traffic control (AERA). In *Proceedings of the American Control Conference*, volume 5, pages 3425 – 3429, 1995.

[30] J. Brueckner. Airport congestion when carriers have market power. *American Economic Review*, 92:1357–1375, 2002.

[31] A.E. Bryson, M.N. Desai, and W.C. Hoffman. Energy-State Approximation in Performance Optimization of Supersonic Aircraft. *Journal of Aircraft*, 6(6):481–488, 1969.

[32] V. Bui, V. V. Pham, A. W. Iorio, J. Tang, S. Alam, and H. A. Abbass. Bio-inspired robotics for air traffic weather information management. *Transactions of the Institute of Measurement and Control*, 2010.

[33] A.J.M. Chakravarty. Time-responsive flight optimization system. US Patents, 1995.

[34] P.R. Chandler, S. Rasumussen, and M. Pachter. UAV cooperative path planning. In *AIAA Guidance, Navigation, Control Conference*, 2000.

[35] K.Y. Chen, P. A. Lindsay, P. J. Robinson, and H. A. Abbass. A Hierarchical Conflict Resolution Method for Multi-Agent Path Planning. In *Proceedings*

of the Eleventh conference on Congress on Evolutionary Computation, pages 1169–1176, 2009.

[36] C. A. C. Coello. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1):28–36, 2006.

[37] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256 – 279, 2004.

[38] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. MIT University Press, Cambridge, 2001.

[39] G. Csernák. *EuroScope Users Guide*, version 3.0a edition, 2008.

[40] K. Culligan, M. Valenti, Y. Kuwata, and J.P. How. Three-dimensional flight experiments using on-line mixed-integer linear programming trajectory optimization. In *Proc. 2007 American Control Conf.*, pages 5322–5327, 2007.

[41] A. d'Aspremont, D. Sohier, A. Nilim, L. El Ghaoui, and V. Duong. Optimal path planning for air traffic flow management under stochastic weather and capacity constraints. In *Proceedings of the 4th IEEE International Conference on Research, Innovation and Vision for the Future, RIVF06*, pages 1–6, 2006.

[42] K. Deb. *Multiobjective Optimization using Evolutionary Algorithms*. John Wiley and Son Ltd, New York, 2001.

[43] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. *Parallel Problem Solving from Nature (PPSN VI)*, chapter A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, pages 849 – 858. Springer, Berlin, 2000.

[44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[45] R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM*, 32(3):505–536, 1985.

[46] J. Doebbler, P. Gesting, and J. Valasek. Real-Time Path Planning and Terrain Obstacle Avoidance for General Aviation Aircraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1–12, Aug 2005.

[47] G. Dowek, C. Mu noz, and A. Geser. Tactical Conflict Detection and Resolution in 3-D Airspace. Technical Report NASA/CR-2001-210853 ICASE Report No. 2001-7, ICASE-NASA Langley, ICASE Mail Stop 132C, NASA Langley Research Center, Hampton VA 23681-2199, USA, Apr 2001.

[48] G. Dowek, C. Mu noz, and V. A. Carre no. Provably safe coordinated strategy for distributed conflict resolution. In *Proceedings of the AIAA Guidance Navigation and Control Conference and Exhibit*, 2005.

[49] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497 – 516, 1957.

[50] D. DuBois and G.C. Paynter. Fuel Flow method 2 for Estimating Aircraft Emissions. Technical Report SAE Technical Paper series-2006-01-198, The Boeing Company, Washington DC, 2006.

[51] V. Duong, E. Hoffman, and J-P. Nicoleon. Initial Results of Investigation into Autonomous Aircraft Concept. In *Proceedings of 1st FAA/Eurocontrol ATM R&D Seminar*.

[52] eDEP project team. Early Demonstration & Evaluation Platform. `http://www.eurocontrol.fr/projects/edep/`, 2007. Accessed 16/02/2012.

[53] A. E. Eiben, P. E. Raué, and Zs. Ruttkay. Genetic algorithms with multiparent recombination. In *The Third Conference on Parallel Problem Solving from Nature*, pages 78 – 87, 1994.

[54] EPISODE. Grossary. http://www.episode3.aero/Glossary, 2012. Accessed 05/03/2012.

[55] H. Erzberger. Transforming the NAS: The Next Generation Air Traffic Control System. In *Proceedings of the International Congress of the Aeronautical Sciences (ICAS)*, Yokohama, Japan, 2004.

[56] Eurocontrol. EATM glossary of terms. Technical report, Eurocontrol, 2004.

[57] Eurocontrol. Digest Annual 2006: Delays to Air Transport in Europe. Technical report, Eurocontrol, Brussels, Belgium, 2006.

[58] Eurocontrol. The Free Route Airspace Concept. http://www.eurocontrol.int/news/flight-plan-processing-enters-free-route-airspace-era, 2009. Accessed 07/12/2011.

[59] C. J. Eyers, D. Addleton, K. Atkinson, M. J. Broomhead, R. Christou, T. Elliff, R. Falk, I. Gee, D. S. Lee, C. Marizy, S. Michot, J. Middel, P. Newton, P. Norman, Plohr M., D. Raper, and N. Stanciou. AERO2K Global Aviation Emissions Inventories for 2002 and 2025. Technical Report QinetiQ/04/01113, QinetiQ, UK, 2004.

[60] FAA. Flight Management System (FMS) Instrument Procedures Development. Technical Report 8260.40B, Federal Aviation Administration, 1998.

[61] FAA. System for assessing Aviations Global Emissions (SAGE), Version 1.5. Technical Report FAA-AEE-2005-01, Office of Environment and Energy (AEE-300), Federal Aviation Administration, 800 Independence Ave., SW, Washington DC 20591, USA, 2005.

[62] FAA. System for assessing Aviations Global Emissions (SAGE), Version 1.5, Global Aviation Emissions Inventories for 2000 through 2004. Technical Report FAA-EE-2005-02, Office of Environment and Energy (AEE-300), Federal Aviation Administration, 800 Independence Ave., SW, Washington DC 20591, USA, 2005.

[63] FAA. FAA's NextGen Implementation Plan. Technical report, Federal Aviation Administration, 2011.

[64] FAA-Eurocontrol. White Paper : Common Trajectory Prediction-Related Terminology. Technical report, FAA/Eurocontrol COOPERATIVE R&D, 2004.

[65] FAA-Eurocontrol. Common TP Structure and Terminology in support of SESAR & NextGen. Technical report, FAA/Eurocontrol COOPERATIVE R&D, 2010.

[66] R. W. Farebrother. *Fitting Linear Relationships: A History of the Calculus of Observations 1750-1900*. Springer, New York, 1st edition, 1999.

[67] D. E. Ferguson. Fibonaccian searching. *Communications of the ACM*, 3(12), 1960.

[68] C. Fonseca and P. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *The Fifth International Conference on Genetic Algorithms*, pages 416 – 423, San Mateo, California, 1993. Morgan Kauffman Publishers.

[69] J. M. Garrell, E. Golobardes, E. E. Bernadó-Mansilla, and X. Llorà. Automatic Diagnosis with Genetic Algorithms and Case-Based Reasoning. *Artificial Intelligence in Engineering*, 13(4), 1999.

[70] D. Godbole, T. Samad, and V. Gopal. Active Multi-Model Control for Dynamic Maneuver Optimization of Unmanned Air Vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1257–1262, Apr 2000.

[71] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Inc, 1989.

[72] P. Goodwin and G. Wright. *Decision Analysis for Management Judgment*. Wiley, 3rd edition, 2004.

[73] B. Graham and C. Guyer. Environmental Sustainability, Airport Capacity, and European Air Transport Liberalization: Irreconcilable Goals? *Journal of Transport Geography*, 7(3):165–180, 1999.

[74] G. Gurney and J.A. Skiera. *Private pilot's handbook of weather*. Aero Publishers (Los Angeles), 1964.

[75] O. J. Hadaller and A. M. Momenthy. The Characteristics of Future Fuels. In *Boeing publication D6-54940, Seattle*, 1989.

[76] P. Hagelauer and F. Mora-Camino. A soft dynamic programming approach for on-line aircraft 4D-trajectory optimization. *European Journal of Operational Research*, 107(1):87–95, 1998.

[77] J.T. Heaton. *Introduction to Neural Networks with Java*. Heaton Research, 2nd edition, 2005.

[78] B. G. Hilburn, M. W. P. Bakker, W. D. Pekela, and R. Parasuraman. The effect of free flight on air traffic controller mental workload, monitoring and system performance. In *Proceedings of the 10th International CEAS Conference on Free Flight*, pages 1–12, Amsterdam, 1997.

[79] J. H. Holland. Processing and processors for schemata. In *Jacks EL, editor. Associative Information Processing*, pages 27–146. New York: Elsevier, 1971.

[80] J. H. Holland and J. S. Reitman. Cognitive systems based on adaptive algorithms. In D. A. Waterman and F. Hayes-Roth, editors, *Pattern directed inference systems*, pages 313–329. New York: Academic, 1978.

[81] J.H. Holland. *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[82] E. Hollnagel. Flight decks and free flight: Where are the system boundaries? *Applied Ergonomics*, 38(4):409 – 416, 2007.

[83] M. H. Holmes. *Introduction to perturbation methods*. Springer, 1995.

[84] J. Horn. *Handbook of Evolutionary Computation*, chapter Multicriteria decision making. Institute of Physics Publishing, 1997.

[85] J. Horn, N. Nafpliotis, and D. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *The First IEEE Conference on Evolutionary Computation*, pages 82 – 87, Piscataway, New Jersey, 1994.

[86] ICAO. *Engine Exhaust Emissions Databank. ICAO Doc. 9646*. International Civil Aviation Organisation, Document Sales Unit, 1000 Sherbrooke Street West, Suite 400, Montreal, Quebec, Canada, 1st edition, 1995.

[87] ICAO. Operation of Aircraft. Technical report, International Civil Aviation Organization, 2001.

[88] ICAO. Guidance on the Use of LTO Emissions Certification Data for the Assessment of Operational Impacts. Technical Report CAEP/6-IP/5, International Civil Aviation Organisation, Committee on Aviation Environmental Protection (CAEP) Working Group 3 (WG3), Alternative Emissions Task Group (AEMTG), Montreal, Quebec, Canada, 2004.

[89] ICAO. Annual Report of the Council. Technical Report Doc 9952, International Civil Aviation Organization, 2010.

[90] T. Inanc, K. Misovec, and R.M. Murray. Nonlinear trajectory generation for unmanned air vehicles with multiple radars. In *43rd IEEE Conference on Decision and Control*, volume 4, pages 3817–3822, 2004.

[91] IPCC. IPCC report 2001, chapter 3: The Carbon Cycle and Atmospheric Carbon Dioxide CO2 concentration trends and budgets, p. 3. Technical report, Intergovernmental Panel on Climate Change (IPCC), 2001.

[92] IPCC. Climate Change 2007 Synthesis Report - Appendix Glossary. Technical report, Intergovernmental Panel on Climate Change (IPCC), 2007.

[93] H. Ishibuchi and T. Murata. Multi-objective genetic local search algorithm. In *IEEE International Conference on Evolutionary Computation*, pages 119–124, 1996.

[94] M. Janic. Aviation and externalities: the accomplishments and problems. *Transportation Research 4D*, 4(3):159–180, 1999.

[95] C. Janikow. *Inductive Learning of Decision Rules in Attribute-Based Examples: a Knowledge-Intensive Genetic Algorithm Approach.* Doctoral dissertation, University of North Carolina, Chapel Hill, Jul 1991.

[96] C. Jansch and M. Paus. Aircraft trajectory optimization with direct collocation using movable gridpoints. In *American Control Conference*, pages 262–267, 1990.

[97] M.R. Jardin. Real-Time Conflict-Free Trajectory Optimization. In *5th USA/Europe ATM 2003 R&D Seminar*, 2003.

[98] M.R. Jardin. *Toward real-time en route air traffic control optimization.* Phd dissertation, Dept. of Aeronautics and Astronautics, Stanford Univ, Stanford, CA, 2003.

[99] K. A. De Jong and W. M. Spears. Learning Concept Classification Rules Using Genetic Algorithms. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 651–656, Sydney, Australia: Morgan Kaufmann, 1991.

[100] JPDO. Concept of Operations for the Next Generation Air Transportation System. Technical Report Version 1.2, Joint Planning and Development Office, 2007.

[101] W. W. Cooper Jr., Dr. E. A. Cherniavsky, J. S. DeArmon, J. G. Foster, Dr. M. J. Mills, Dr. S. C. Mohleji, and F. Z. Zhu. Determination of Minimum Push-Back Time Predictability Needed for Near-Term Departure Scheduling using DEPARTS. In *Proceedings of the Fourth USA/Europe Air Traffic Management*

Seminar ATM-2001, Santa Fe NM, 2001. Eurocontrol and the US Federal Aviation Administration.

[102] R.L. Keeney and H. Raiffa. *Decisions with multiple objectives: Preferences and value tradeoffs*. Cambridge University Press, 1993.

[103] J. J. Kehoe, A. S. Watkins, and R. Lind. A Time-Varying Hybrid Model for Dynamic Motion Planning of an Unmanned Air Vehicle. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Aug 2006.

[104] E. King, Y. Kuwata, M. Alighanbari, L. Bertuccelli, and J. How. Coordination and control experiments on a multi-vehicle testbed. In *Proceedings of American Control Conference*, volume 6, pages 5315–5320, 2004.

[105] J. D. Knowles and D. W. Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Congress on Evolutionary Computation*, pages 98 – 105. IEEE Press, 1999.

[106] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149 – 172, 2000.

[107] D.E. Knuth. *Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison–Wesley, 1997.

[108] A. Kosorukoff. Human-based Genetic Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 5:3464–3469, 2001.

[109] F. Kursawe. *Parallel Problem Solving from Nature*, volume 496/1991, chapter A variant of evolution strategies for vector optimization, pages 193 – 197. Springer, 1991.

[110] Y. Kuwata and J. How. Three dimensional receding horizon control for UAVs. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 16–19, 2004.

[111] Y. Kuwata and J.P. How. Stable trajectory design for highly constrained environments using receding horizon control. In *Proceedings of the American Control Conference*, volume 1, pages 902–907, 2004.

[112] P.F. Lester. *Aviation weather*. Jeppesen Sanderson, 2nd edition, 2004.

[113] X. Llorá, D. E. Goldberg, I. Traus, and E. Bernado-Mansilla. Accuracy, Parsimony, and Generality in Evolutionary Learning Systems via Multiobjective Selection. In *Learning classifier system: Fifth international workshop, IWLCS*, volume 2661, pages 118–142. Springer, 2003.

[114] P. L. Mackeen, H. E. Brooks, and K. L. Elmore. Radar reflectivity-derived thunderstorm parameters applied to storm longevity forecasting. *Weather and forecasting*, 14(2):289–295, 1999.

[115] B. McNally, R. Bach, and W. Chan. Field test evaluation of the CTAS conflict prediction and trial planning capability. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1686–1697, 1998.

[116] A. Melrose. Basic continuous descent approach. In *Airports and Environmental Management*, 2004.

[117] S. Michot, N. Stancioi, C. McMullan, S. Peeters, and S. Carlier. AERO2k Flight Movement Inventory. Technical Report EEC/SEE/2003/005, Eurocontrol Experimental Centre, Bretigny-sur-Orge, France, Dec 2003.

[118] A. Miele. Optimal trajectories and guidance trajectories for aircraft flight through windshears. In *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 737–746, 1990.

[119] A. Miele, T. Wang, and W.W. Melvin. Optimal take-off trajectories in the presence of windshear. *Journal of Optimization Theory and Applications*, 49(1):1–45, 1986.

[120] A. Miele, T. Wang, and W.W. Melvin. Quasi-steady flight to quasi-steady flight transition for abort landing in a windshear: Trajectory optimization

and guidance. *Journal of Optimization Theory and Applications*, 58(2):165–207, 1988.

[121] K. Miettinen. *Nonlinear Multiobjective Optimization.* Boston, USA: Kluwer Academic Publishers, 1999.

[122] M.B. Milam, K. Mushambi, and R.M. Murray. A new computational approach to real-time trajectory generation for constrained mechanical systems. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 1, pages 845–851, 2000.

[123] A. Mille and T. Wang. Acceleration, Gamma, and Theta Guidance Schemes for Abort Landing Trajectories in the Presence of Windshear. *Journal of Guidance, Control, and Dynamics*, 12(6):815–821, 1987.

[124] A. Mille, T. Wang, W. W. Melvin, and C. Y. Tzeng. Transformation Techniques for Minimax Optimal Control Problems and Their Application to Optimal Flight Trajectories in a Windshear: Optimal Abort Landing Trajectories. In *Proceedings of the Tenth Triennial World Congress of IFAC*, pages 131–150, 1987.

[125] A. Mille, T. Wang, C. Y. Tzeng, and W. W. Melvin. Optimization and Guidance of Abort Landing Trajectories in a Windshear. In *AIAA Guidance, Navigation, and Control Conference*, pages 483–509, Monterey, California, CA, United States, 1987.

[126] T. Mitchell. *Machine Learning.* McGraw Hill, 1997.

[127] S. Mittal and K. Deb. Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms. In *IEEE Congress on Evolutionary Computation*, pages 3195–3202, Sep 2007.

[128] R. Murphey, S.Uryasev, and M.Zabarankin. Trajectory optimization in a threat environment. Research Report 2003-9, Dept. of Industrial & Systems Engineering, University of Florida, 2003.

[129] NATCA. Air Traffic Control: By the Numbers. `http://www.natca.org/mediacenter/bythenumbers.msp`, 2010. NATCA – National Air Traffic Controllers Association.

[130] R. Neufville. Management of multi-airport systems: A development strategy. *Journal of Air Transport Management*, 2(2):99–110, 1995.

[131] NGGIC. *Australian Methodology for the Estimation of Greenhouse Gas Emissions and Sinks 2006. Workbook for Transport*. National Greenhouse Gas Inventory Committee (NGGIC), Department of Climate Change, Canberra, Australia, 2007.

[132] N. Nguyen. Singular Arc Time-Optimal Climb Trajectory of Aircraft in a Two-Dimensional Wind Field. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, 2006.

[133] A. Nilim, L. El Ghaoui, and V. Duong. Multi-aircraft routing and traffic flow management under uncertainty. In *5th USA/Europe Air Traffic Management Research and Development Seminar*, pages 23–27, Budapest, Hungary, 2003.

[134] M. Nolan. *Fundamentals of Air Traffic Control*. Brooks Cole, 4th edition, 2004.

[135] A. Nuic. *User Manual for the Base of Aircraft Data (BADA) - Revision 3.6*. Eurocontrol Experimental Centre, Bretigny-sur-Orge, France, 2004.

[136] A. Nuic, P. Chantal, M.G. Iagaru, E. Gallo, F.A. Navarro, and C. Querejeta. Advanced aircraft performance modeling for ATM: enhancements to the BADA model. In *Proceedings of the 24th digital avionics systems conference*, Washington, DC, 2005.

[137] OpenMP Architecture Review Board. *OpenMP C and C++ Application Program Interface*. 2nd edition, 2002.

[138] M. Paglione, I. Bayraktutar, G.N. McDonald, and J. Bronsvoort. Lateral Intent Error's Impact on Aircraft Prediction. *Air Traffic Control Quarterly*, 18(1):29–62, 2010.

[139] J. E. Penner, D. Lister, D. Griggs, D. Docken, and M. MacFarland. *Aviation and the Global Atmosphere*. Cambridge University Press , New York, 1999.

[140] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*, chapter VIII, page 360. Wiley, New York, 1962.

[141] J. Prete, J. Krozel, J. S. B. Mitchell, and J. Kim. Flexible, Performance-based Route Planning for Super-Dense Operations. In *AIAA Guidance, Navigation, and Control Conference*, 2008.

[142] L. Qing, G. Wei, L. Yuping, and S. Chunlin. Aircraft route optimization using genetic algorithms. In *Second International Conference On Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 394–397, 1997.

[143] R.L. Raffard, C.J. Tomlin, and S.P. Boyd. Distributed optimization for cooperative agents: application to formation flight. In *43rd IEEE Conference on Decision and Control*, volume 3, pages 2453–2459, 2004.

[144] N.S. Rao, N.L. Phillips, S.J. Fu, and N.M. Conrardy. Horizontal plane trajectory optimization for threat avoidance and waypoint rendezvous. In *Proceedings of the IEEE National Aerospace and Electronics Conference*, volume 2, pages 580–583, 1990.

[145] A. Richards. *Trajectory optimization using mixed-integer linear programming*. PhD thesis, Massachusetts Institute of Technology, Jun 2002.

[146] A. Richards, J. Bellingham, M. Tillerson, and J. How. Coordination and control of multiple UAVs. In *Proceedings of the AIAA guidance, navigation, and control conference*, Monterey, CA, 2002.

[147] A. Richards and J. How. Decentralized model predictive control of cooperating UAVs. In *43rd IEEE Conference on Decision and Control*, volume 4, pages 4286–4291, 2004.

[148] A. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the American Control Conference*, volume 3, pages 1936–1941, May 2002.

[149] A. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the American Control Conference*, volume 3, pages 1936–1941, 2002.

[150] A. Richards, Y. Kuwata, and J. How. Experimental demonstrations of real-time MILP control. In *Proceeding of the AIAA Guidance, Navigation, and Control Conference*, 2003.

[151] O. Richetta. *Ground Holding Strategies for Air Trafie Control Under Uncertainty*. PhD thesis, Massachusetta Institute of Technobgy, 1991.

[152] O. Richetta and A. R. Odoni. Dynamic Solution to the Ground-Holding Policy Problem in Air Traffic Control. *Transportation Research A*, 28:169 – 185, 1994.

[153] O. Richetta and A.R. Odoni. Solving Optimally the Static Ground-Holding Policy Problem in Air Traffic Control. *Transportation Science*, 27(13):228 – 238, 1993.

[154] S.J. Russell, P. Norvig, J.F. Canny, J.M. Malik, and D.D. Edwards. *Artificial intelligence: a modern approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[155] E.S. Rutowski. Energy Approach to the General Aircraft Performance Problem. *Journal of the Aerospace Sciences*, 21(3):187–195, 1954.

[156] J.J. Ruz, O. Arévalo, JM de la Cruz, and G. Pajares. Using MILP for UAVs trajectory optimization under radar detection risk. In *IEEE Conference on Emerging Technologies and Factory Automation*, pages 957–960, 2006.

[157] G. Saini and SN Balakrishnan. Adaptive critic based neurocontroller for autolanding of aircrafts. In *Proceedings of the American Control Conference*, volume 2, pages 1081–1085, 1997.

[158] M. Schaefer. *Methodologies for Aviation Emission Calculation – A comparison of alternative approaches towards 4D global inventories*. PhD thesis, Berlin University of Technology In cooperation with the German Aerospace Center (DLR), 2006.

[159] J. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Genetic Algorithms and their Applications*, pages 93 – 100, Hillsdale, New Jersey, 1985.

[160] J.D. Scheelhaase and W.G. Grimme. Emissions trading for international aviation an estimation of the economic impact on selected European airlines. *Journal of Air Transport Management*, 13(5):253 – 263, 2007.

[161] T. Schouwenaars, J. How, and E. Feron. Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pages 6820–6825, 2004.

[162] J.M. Schreur. B737 flight management computer flight plan trajectory computation and analysis. In *Proceedings of the American Control Conference*, volume 5, pages 3419 – 3424, Seattle, WA , USA, 1995.

[163] U. Schumann. Aircraft Emissions. In *Encyclopedia of Global Environmental Change*, volume 3, pages 178–186. Wiley, 2002.

[164] SESAR. SESAR Definition Phase: The Concept of Operations at a glance. Technical Report DLT 2.2.2/D3, SESAR - Single European Sky ATM Research, 2007.

[165] SESAR. European Air Traffic Management Master Plan. Technical report, SESAR, 2009.

[166] S.C. Shapiro. *Encyclopedia of artificial intelligence*. New York: John Wiley & Sons, 2nd edition, 1992.

[167] A. Sharma and A. Chaturvedi. Gradient Descent Feed Forward Neural Networks for Forecasting the Trajectories. *International Journal of Advanced Science and Technology*, 34, 2011.

[168] G. Shepherd, B. Byron, M. Mrdak, and K. Bills. WAYPOINT 2007 Airservices Australia Air Navigation Service Provision: Moving towards 2020. Opening Address, 2007.

[169] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman and Hall, 1st edition, 1986.

[170] R.E. Smith, BA Dike, RK Mehra, B. Ravichandran, and A. El-Fallah. Classifier systems in combat: two-sided learning of maneuvers for advanced fighter aircraft. *Computer Methods in Applied Mechanics and Engineering*, 186(2–4):421–437, 2000.

[171] S. F. Smith. Flexible Learning of Problem Solving Heuristics through Adaptive Search. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, volume 1, pages 422–425, 1983.

[172] J.A. Sorensen, S.A. Morello, and H. Erzberger. Application of trajectory optimization principles to minimize aircraft operating costs. In *18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 415–421, 1979.

[173] B. Sridhar, N.Y. Chen, H. K. Ng, and F. Linke. Design of Aircraft Trajectories based on Trade-offs between Emission Sources. In *Ninth USA/Europe Air Traffic Management Research and Development Seminar*, 2011.

[174] M. Srinivas and L.M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 24(4):656–667, 1994.

[175] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221 – 248, 1994.

[176] R. Sroul. *Programming for Mathematicians*. Springer, Berlin, 1st edition, 2000.

[177] A. P. Sturman and N. J. Tapper. *The Weather and Climate of Australia and New Zealand*. Oxford University Press, Melbourne, Australia, 2nd edition, 2005.

[178] D. J. Jr. Sutkus, S. L. Baughcum, and D. P. DuBois. Scheduled Civil Aircraft Emission Inventories for 1999: Database Development and Analysis. Technical report, NASA, Washington, D.C., 2001.

[179] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[180] M. N. S. Swamy and K. Thulasiraman. *Graphs, Networks and Algorithms*. Wiley, 1st edition, 1980.

[181] K.S. Tang, K.F. Man, S. Kwong, and Q. He. Genetic algorithms and their applications. *Signal Process Magazine*, 13(6):22 – 37, 1996.

[182] M. Terrab. Ground Holding Strategies in Air Traffic Control. Technical Report No. 196, Operations Research Center, MIT, Cambridge, MA, 1990.

[183] M. Terrab and A. R. Odoni. Strategic Flow Management for Air Traffic Control. *Operations Research*, 41(1):138–152, 1993.

[184] M. Terrab and S. Paulose. Dynamic Strategic and Tactical Air Traffic Flow Control. Rpi technical report, 1993.

[185] C.-K. Ting. On the Mean Convergence Time of Multi-parent Genetic Algorithms Without Selection. *Advances in Artificial Life*, 3630/2005:403 – 412, 2005.

[186] A. Tomer and R. Puentes. Expect Delays: An Analysis of Air Travel Trends in the United States. Technical report, Brookings Institution Metropolitan Program, 2009.

[187] Transport Canada. Flight 2010: A strategic plan for civil aviation. Technical Report TP 14469, Minister of Transport, Ottawa, Canada, 2006.

[188] TSC. Direct-To (D2) Overview Description. Technical Report NAS2-98005, TSC - Titan Systems Corporation, Billerica, Massachusetts, USA, 2002.

[189] S. Twigg, A. Calise, and E. Johnson. On-line trajectory optimization for autonomous air vehicles. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2003.

[190] K. Virtanen, H. Ehtamo, T. Raivio, and R.P. Hamalainen. VIATO-visual interactive aircraft trajectory optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 29(3):409–421, 1999.

[191] K. Virtanen, T. Raivio, and R.P. Hamalainen. Modeling pilot's sequential maneuvering decisions by a multistage influence diagram. *Journal of Guidance Control and Dynamics*, 27(4):665–677, 2004.

[192] H.G. Visser. Generic and site-specific criteria in the optimization of noise abatement trajectories. *Transportation Research Part D: Transport and Environment*, 10(5):405–419, 2005.

[193] P.B. Vranas, D.J. Bertsimas, and A.R. Odoni. The Multi-Airport Ground-Holding Problem in Air Trafic Control. *Operation Research*, 42(2):249 – 261, 1994.

[194] I. Waitz, J. Townsend, J. Cutcher-Gershenfeld, E. Greitzer, and J. Kerrebrock. Report to the United States Congress: Aviation and the Environment, A National Vision, Framework for Goals and Recommended Actions. Partnership for air transportation noise and emissions reduction, Massachusetts Institute of Technology, Cambridge, MA, 2004.

[195] A. Warren, K. Tong, B. Manage, and W. Seattle. Development of continuous descent approach concepts for noise abatement. In *21st AIAA/IEEE Digital Avionics Systems Conference*, volume 1, 2002.

[196] M.C. Weinstein and H.V. Fineberg. *Clinical decision analysis*. Saunders, Philadelphia, 1980.

[197] H. Wesoky, U. Schumann, and D. Wurzel. Lingering Uncertainty about Aviation's Impact Addressed by Growing Body of Scientific Data. *ICAO Journal*, 51:11–14, 1996.

[198] D. Von Winterfeldt and W. Edwards. *Decision analysis and behavioral research*, volume 1. New York: Cambridge University Press Cambridge, 1986.

[199] G. Yang and V. Kapila. Optimal Path Planning for Unmanned Air Vehicles with Kinematic and Tactical Constraints. In *41st IEEE Conference on Decision and Control*, volume 2, pages 1301 – 1306, Dec 2002.

[200] N. Yokoyama and S. Suzuki. Modified Genetic Algorithm for Constrained Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 28(1):139–144, 2005.

[201] M. Zabarankin, S. Uryasev, and R. Murphey. Aircraft routing under the risk of detection. *Naval Research Logistics (NRL)*, 53(8):728–747, 2006.

[202] M. Zabarankin, S. Uryasev, and P. Pardalos. Optimal risk path algorithms. *Cooperative Control and Optimization*, 66:273 – 298, 2002.

[203] N.R. Zagalsky, R.P Irons, and R.L. Schultz. Energy State Approximation and Minimum-Fuel Fixed-Range Trajectories. *Journal of Aircraft*, 8(6):488–490, 1971.

[204] J. Zhang, H. S.-H. Chung, and W.-L. Lo. Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms. *IEEE Transactions on Evolutionary Computation*, 11(3):326 – 335, 2007.

[205] E. Zitzler, M. Laumanns, and S. Bleuler. A Tutorial on Evolutionary Multi-objective Optimization. *Metaheuristics for Multiobjective Optimisation*, 535:3 – 37, 2004.

[206] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems*, Athens, Greece, 2001.

[207] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257 – 271, 1999.