On Modeling and Optimisation of Air Traffic Flow Management Problem with En-route Capacities

UNIVERSITY OF KWAZULU-NATAL

Alex Somto Arinze ALOCHUKWU
(215036740)

A thesis submitted in partial fulfilment of the academic requirements
for the award of the degree of Master of Science

in the

School of Mathematics, Statistics and Computer Science
University of KwaZulu-Natal
Durban, South Africa

February, 2016

UNIVERSITY OF KWAZULU-NATAL

COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

DECLARATION

This research was conducted at the University of KwaZulu-Natal under the supervision of Prof. A. O. Adewumi. I hereby declare that all material incorporated in this thesis is my own original work except where acknowledgement is made by name or in form of a reference. The work contained herein has not been submitted in part or whole for a degree at any other university.

Signed:

____

Alex Somto Arinze Alochukwu

Date: February, 2016

As the candidate's supervisor, I have approved the dissertation for submission.

Signed:

____

Prof. A. O. Adewumi
Date: February, 2016

As the candidate's co-supervisor, I have approved the dissertation for submission.

Signed:

____

Dr Micheal O. Olusanya
Date: February, 2016

UNIVERSITY OF KWAZULU-NATAL

COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

DECLARATION 1- PLAGIARISM

I, Alex Somto Arinze Alochukwu, declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons or researchers.
4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other persons or researchers. Where other written sources have been quoted, then:
   (a) Their words have been re-written but the general information attributed to them has been referenced.
   (b) Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.
5. This thesis does not contain text, graphics or tables copied and pasted from the internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed:

____

Alex Somto Arinze Alochukwu

Date: February, 2016

Dedication

To

God Almighty; In Whom I live, move and have my being

My lovely mother: Lily Nwabogo Nwankwo

My future wife and unborn children

My wonderful nuclear and extended family

My lovely friends, classmates and colleagues

Acknowledgements

I am forever grateful to the only wise and true God, whose love, mercy, tender care and faithfulness has brought me this far and will continue to lead me through till the end. Thank you Lord for the gift of life, perfect health, good sight, favour, inspiration, wisdom, knowledge, understanding, guidance and protection throughout my programme.

My heartfelt appreciation goes to my humble and indefatigable supervisors, Prof A. O. Adewumi and Dr M. O. Olusanya, for their motivation, encouragement, moral support, kind assistance, guidance and interest in the progress of my work through this period of research. Many thanks to Professor Montaz Ali, Dr Ian Campbell, Professor Guglielmo, Dr Antonio and Dr Laureano for their kind assistance, commitment and meaningful contributions to the success of this work. I sincerely appreciate all your efforts especially the time you created out of your busy schedules to attend to all my queries regarding the basic concepts and the resource materials provided. In fact, my meeting with all of you was divinely arranged and I'm deeply indebted to you all. A very big thank you to Princess Alice Sibanda for her kind assistance in proofreading my work. Accept my heartfelt gratitude.

To my parents, The Late Ericmoore Azuka and Mrs Lily Nwabogo Nwankwo, who went through insults and sacrificed all they had just to see that their children are comfortable, I say a very big thank you. I pray that God will bless and keep my dearest mother in good health to eat the fruit of her labour.

To my immediate family members, Chinenye, Eloka, Godswill, Amara, my granny, my godmother, aunts, cousins, uncles, in-laws, friends and relatives, I say a big thank you for the love, unwavering support and encouragement throughout my academic journey. You have contributed immensely to my success story.

Special thanks to the African Institute for Mathematical Sciences (AIMS) Ghana, School of Mathematics, Statistics and Computer Science, College of Agriculture, Engineering and Sciences, for the financial supports and wonderful opportunity granted unto me to further my studies. Most importantly, for providing an enabling and conducive learning environment. I must fail to thank Mrs Shereen Marimuthu, Mrs Charmaine Magwaza, Mrs Selvie Moodley, Mr Soren Greenwoods, all other academic and administrative staff and students of the school for their warm reception, assistance and cooperation during this period of research. It was indeed a great pleasure working with you all.

Thanks to Dr C. R. Chikwendu, Mr Nnanwa Chimezie, Prof Eric Ofoedu, Prof Chika Moore, Mr Bernado Ugwu, Chinenye Ezeamama and other staff and students of the Department of Mathematics, Nnamdi Azikiwe University Awka, for their positive impact and kind assistance all these years.

My immense gratitude goes to my priests, pastors, spiritual fathers, to mention but a few, Ven & Mrs A. I. Umeh, Canon & Mrs Izuu Obiekwe, Revd & Mrs Mike Okoh, Revd & Mrs I. I. Okafor, Revd & Mrs Chibu Abone, Revd & Mrs Izuu Ndubuisi, Revd & Mrs Nzegwu, Pastor Joseph Adesina, Pastor Martins Akugbe, and all members of DLBC, KZN province, for their prayers and support.

I would also like to appreciate my research (Optimization and Applied Artificial Intelligence Research) group members, Mr Olawale, Ayo, Val, Peter & Seun, Peter, Ebenezer, Steve, Raphael for their meaningful contribution and the quality time we spent sharing ideas together; my office colleagues, Mafunda, Lulama, Banele, Keith, for the warm company and finally Leena, Nerson, Sash, Precious, Phindile and other CAES postgraduate network team members, for the beautiful moments we have shared together. It was indeed a memorable experience with all of you as I enjoyed every bit of the moments we have shared.

I should not fail to thank these persons: Prof E. K. Essel, Dr Prince Ossei, Dr Ulrich Krahmer, Dr Alistair Wallis, Sarah Ossei, Dr Mewomo, Mummy Olaoluwa Adewumi, Mummy Ruth Olusanya, Dr & Mrs Adebayo, Dr & Mrs Adeleke for their unalloyed support. Thanks to my dearest friends, Chidimma, Emelie, Sunny, Ikenna, Ord Jossy & Juliet, Ord Iyke, Gentle, Chidi, Onyekachukwu, Chinaza, Onyedika, Nonso, Tochukwu, Nkechi, Ceejay, Arinze, Izuu, Emeka, Nnaemeka, Femi, Angela, Benji, Beltako, Prosper, Shola, Bayo, Muyiwa, Vicky, Majee & Ifuu, Chinedu, Jerry, Soji, Komi, Larweh & Mary, Langa, Samson and numerous others for their love, care and support.

Finally, I say a big thank you to all whose works were cited in this thesis, especially Prof A. R. Odoni, Prof D. Bertsimas, Prof Agustin, Hriskesh Ganu and many others who in one way or the other contributed directly or indirectly to the success of this work. May God bless you all.

Somchuks Alex

February, 2016

Abstract

The air transportation industry in the past ten years witnessed an upsurge with the number of passengers swelling exponentially. This development has seen a high demand in airport and airspace usage, which consequently has an enormous strain on the aviation industry of a given country. Although increase in airport capacity would be logical to meet this demand, factors such as poor weather conditions and other unforeseen ones have made it difficult if not impossible to do such. In fact there is a high probability of capacity reduction in most of the airports and air sectors within these regions. It is no surprise therefore that, most countries experience congestion almost on a daily basis. Congestion interrupts activities in the air transportation network and this has dire consequences on the air traffic control system as well as the nation's economy due to the significant costs incurred by airlines and passengers.

This is against a background where most air traffic managers are met with the challenge of finding optimal scheduling strategies that can minimise delay costs. Current practices and research has shown that there is a high possibility of reducing the effects of congestion problems on the air traffic control system as well as the total delay costs incurred to the nearest minimum through an optimal control of flights. Optimal control of these flights can either be achieved by assigning ground holding delays or air borne delays together with any other control actions to mitigate congestion. This exposes a need for adequate air traffic flow management given that it plays a crucial role in alleviating delay costs.

Air Traffic Flow Management (ATFM) is defined as a set of strategic processes that reduce air traffic delays and congestion problems. More precisely, it is the regulation of air traffic in such a way that the available airport and airspace capacity are utilised efficiently without been exceeded when handling traffic. The problem of managing air traffic so as to ensure efficient and safe flow of aircraft throughout the airspace is often referred to as the Air Traffic Flow Management Problem (ATFMP).

This thesis provides a detailed insight on the ATFMP wherein the existing approaches, methodologies and optimisation techniques that have been (and continue to be) used to address the ATFMP were critically examined. Particular attention to optimisation models on airport capacity and airspace allocation were also discussed extensively as they depict what is obtainable in the air transportation system. Furthermore, the thesis attempted a comprehensive and, up-to-date review which extensively fed off literature on ATFMP. The instances in this literature were mainly derived from North America, Europe and Africa.

Having reviewed the current ATFM practices and existing optimisation models and approaches for solving the ATFMP, the generalised basic model was extended to account for additional modeling variations. Furthermore, deterministic integer programming formulations were developed for reducing the air traffic delays and congestion problems based

on the sector and path-based approaches already proposed for incorporating rerouting options into the basic ATFMP model. The formulation does not only takes into account all the flight phases but it also solves for optimal synthesis of other flow management activities including rerouting decisions, flight cancellation and penalisation. The claims from the basic ATFMP model was validated on artificially constructed datasets and generated instances. The computational performance of the basic and modified ATFMP reveals that the resulting solutions are completely integral, and an optimal solution can be obtained within the shortest possible computational time. Thereby, affirming the fact that these models can be used in effective decision making and efficient management of the air traffic flow.

Contents

List of Abbreviations

ACO	**A**nt **C**olony **O**ptimisation
AD	**A**irborne **D**elay
ADL	**A**irport **D**emand **L**ist
AFP	**A**irspace **F**low **P**rogramme
AFT	**A**irport **F**low **T**ool
AH	**A**irborne **H**olding
AMPL	**A** **M**athematical **P**rogramming **L**anguage
AMT	**A**irspace **M**anagement **T**ool
ARTCC	**A**ir **R**oute **T**raffic **C**ontrol **C**enters
ATC	**A**ir **T**raffic **C**ontrol
ATCSCC	**A**ir **T**raffic **C**ontrol **S**ystem **C**ommand **C**enter
ATFM	**A**ir **T**raffic **F**low **M**anagement
ATFMP	**A**ir **T**raffic **F**low **M**anagement **P**roblem
ATFMRP	**A**ir **T**raffic **F**low **M**anagement **R**erouting **P**roblem
ATM	**A**ir **T**raffic **M**anagement
ATNS	**A**ir **T**raffic **a**nd **N**avigation **S**ervices
B&B	**B**ranch **a**nd **B**ound
BATFMP	**B**asic **A**ir **T**raffic **F**low **M**anagement **P**roblem
CAMU	**C**entral **A**irspace **M**anagement **Unit**
CASA	**C**omputer **A**ssisted **S**lot **A**llocation
CDM	**C**ollaborative **D**ecision **M**aking
CPLEX	**I**BM **I**LOG CPLEX **O**ptimisation **S**tudio
DEM	**D**eterministic **E**quivalent **M**odel
EU	**E**urpoean
EUROCAT	**T**hales **A**ir **T**raffic **C**ontrol **a**nd **M**anagement **S**olution

FAA	**F**ederal **A**viation **A**dministration
FCA	**F**low **C**onstrained **A**rea
FLOWCAT	**T**hales **A**ir **T**raffic **F**low **M**anagement **S**ystem
FPFS	**F**irst **P**lanned **F**irst **S**erved
GA	**G**enetic **A**lgorithm
GD	**G**round **D**elay
GDP	**G**ross **D**omestic **P**roduct
GDP	**G**round **D**elay **P**rogrammes
GH	**G**round **H**olding
GLPK	**G**NU **L**inear **P**rogramming **K**it
GLPSOL	**G**eneral **L**inear **P**rogramming **S**olver
GMPL	**G**NU **M**ath **P**rog **L**anguage
GNU	**F**ree **S**oftware **O**perating **S**ystem
GRASP	**G**reedy **R**andomised **A**daptive **S**earch **P**rocedure
HC	**H**ill **C**limbing
IATA	**I**nternational **A**ir **T**ransportation **A**uthority
IBM	**I**nternational **B**usiness **M**achines
ILOG	**I**nternational **S**oftware **C**ompany
IP	**I**nteger **P**rogramming
LP	**L**inear **P**rogramming
MAGHP	**M**ulti **A**irport **G**round **H**olding **P**roblem
MAS	**M**ulti **A**irport **S**cheduler
METRON	**M**etron **A**viation
MINIT	**M**inutes **I**n **T**rail
MIP	**M**ixed **I**nteger **P**rogramming
MIT	**M**iles **I**n **T**rail
NAC	**N**on **A**nticipativity **C**onstraint

NAS	**N**ational **A**irspace **S**ystem
NP	**N**on **D**eterministic **P**olynomial **T**ime
OAG	**O**fficial **A**irline **G**uide
O-D	**O**rigin **t**o **D**estination
PSO	**P**article **S**warm **O**ptimisation
SA	**S**imulated **A**nnealing
SAA	**S**ample **A**verage **A**pproximation
SAGHP	**S**ingle **A**irport **G**round **H**olding **P**roblem
SATFM	**S**tochastic **A**ir **T**raffic **F**low **M**anagement
SELFMAT	**S**elf **M**anaged **A**ir **T**raffic
TD	**T**otal **D**elay
TS	**T**abu **S**earch
TMI	**T**raffic **M**anagement **I**nitiative
TRACON	**T**erminal **R**adar **A**pproach **C**ontrol **F**acilities
USA	**U**nited **S**tates **o**f **A**merica
ZAR	**S**outh **A**fican **R**and

List of Figures

List of Tables

List of Included Articles

Article under review for peer-review journal

1. Alex Somto Alochukwu and Aderemi Oluyinka Adewumi *On Modeling and Optimisation of Air Traffic Flow Management Problem: A Review.*

Article(s) in preparation for submission to journals

1. Alex Somto Alochukwu and Aderemi Oluyinka Adewumi: *On Deterministic Integer Programming Models for the Air Traffic Flow Management Rerouting Problem.*
2. Alex Somto Alochukwu and Aderemi Oluyinka Adewumi: *On Integer Programming Models for reducing congestion and air traffic delays.*

Chapter 1

Introduction

This chapter is an overview of air traffic flow management, which forms the basis of this study. Within this introductory chapter, the objectives, problem statement, significance and scope of the study as well as the factors that motivated this research are also discussed.

1.1 Background to the Study

The air transportation industry is a key sector in most economies. It is the world's most important service industry owing to the catalytic and significant role it plays in most global economic activities. As a case in point, Oxford Economics in the 2011 report on the economic benefits of air transport in South Africa noted that the country's aviation sector contributes ZAR 50.9 billion (2.1%) to its Gross Domestic Product (GDP). This is through the *contributions made directly from the aviation sector; indirectly through the aviation sector's supply chain as well; through the induced effects of spending by the employees of the aviation sector* (Oxford Economics, 2011). It was also reported that the overall contribution stood at ZAR 74.3 billion (3.1% of South African GDP) if the additional 'catalytic' benefits through tourism were added. Other than that, the sector also creates and supports 227 000 direct jobs with another 116 000 people employed through *tourism effects of aviation* (Department of Transport, 2015; Oxford Economics, 2011).

Air traffic in most countries, as well as the aviation industry, have witnessed a rapid growth over the last decade. This upsurge can be attributed to multifarious factors, chief amongst them, an increased demand for airport and airspace resources due to the exponential increase in the number of users. Thus, flight delays and congestion have become a common phenomenon owing to the capacity reductions in the system resources which are often associated with peak travel times or hours, bad weather conditions and other unforeseen factors. As pointed out by Roosens (2008), the so-called congestion in air transportation occurs when the demand for infrastructure exceeds capacity. In his words, "congestion occurs when there is overcrowded airspace or airport and this causes delays in the travel time" (Bertsimas and Odoni, 1997; Roosens, 2008). However, this does not necessarily imply that all delays are caused by congestion, there are other contributing factors such as adverse weather conditions, en route problems, airline companies, airport and security procedures. Departure delays and crowded airspace serves as indicators of airport and en-route congestion problems (Agustín et al., 2009; Bertsimas et al., 2008; Roosens, 2008).

Statistics shows that in the year 2007, *approximately one out of every four flights in the United States was either delayed or cancelled* (Agustín et al., 2009). It was also reported in the same year that approximately 11% of flights were delayed in Europe. These delays have been estimated to have a direct impact on the economy annually. The Joint Economic Committee, United States Senate (2008) estimated that *the domestic flight or system delays cost passengers, airlines,* will cost the US economy nothing less than 40 billion dollars (Vossen et al., 2012). These delays can be weather-related or attributable to the traffic volume (Agustín et al., 2009; Vossen et al., 2012).

Therefore, the steady growth of the aviation sector over the years has indeed placed excessive demand on the air transportation system which tends to have a serious impact on a nation's economy due to the significant costs incurred by airlines and passengers. Moreover, it has also been predicted that the increase in air traffic and demand for air transportation will continue at an average growth rate of 4.7% over the next 20 years and this continued growth poses a great threat to air traffic control and airport systems which are currently being operated at full capacity. The capacity of the air traffic control systems and that of airports is also greatly affected by the recent practice of "hub and spoke" systems[1] by most airlines, particularly those in the United States, Western Europe and Africa. Thus, in order to meet up with the imminent future demand for air transportation as well as accommodating the projected growth in air traffic, there is need for improvement in air traffic management processes and traffic flow management procedures alongside capacity enhancement of the system's resources as incremental changes in the physical capacity and current operations alone may not be able to meet up with the future demand for air transportation (Odoni, 1987; Sridhar et al., 2008; Vossen et al., 2012).

Air Traffic Management (ATM) is a broader term that is used to represent the overall collection of the air traffic management processes (Vossen et al., 2012). More precisely, it is a composite of services that are taken to ensure safe, efficient and expeditious movement of aircraft in the airspace. It comprises of two basic components namely Air Traffic Control (ATC) and Air Traffic Flow Management (ATFM). ATC refers to those processes that provide tactical separation services for conflict detection and avoidance while ATFM is a set of strategic processes that mitigate delay costs and congestion problems. Most authorities have defined it as "the regulation of air traffic in order to avoid exceeding airport and airspace capacity while making effective use of available capacity when handling traffic" (Bertsimas and Patterson, 1998; Odoni, 1987; Sridhar et al., 2008; Vossen et al., 2012).

For clarity's sake, ATC generally controls individual aircraft and is usually performed by human controllers with the aim of maintaining separation between aircraft while moving traffic as quickly as possible and presenting the traffic in an orderly manner to the next sector. ATC actions are particularly more tactical in nature and primarily address imme-

[1] using more than one airport as operational center (hubs).

diate safety concerns of airborne flights. On the other hand, ATFM is strategic in nature with the objective of matching the demand for air transportation with the capacity of the system in order to ensure a safe and efficient flow of aircraft through the airspace. Moreover, The processes of ATFM are geared towards finding a dynamic equilibrium between flight demands and airport capacity for safer operations such that flight overloading is prevented through adjusted traffic flow aggregation, thereby optimizing scarce capacity resources. Thus, protecting the ATC system from overloading and at the same time reducing the tasks of traffic controllers to a manageable level. In fact, ATFM plays a vital role in maintaining system safety and addressing system efficiency. Above all, ATFM in the short term tries to mitigate congestion problems that arise as a result of weather disturbances or unpredictable disruptions, and if at all, there must be delay, ATFM procedures is efficiently applied within the shortest possible times to reduce the impact on airspace user (Agustín et al., 2009; Vossen et al., 2012).

System disruptions, bad weather conditions, equipment failures and sudden increase in demand, are a major challenge for ATFM as they are highly unpredictable and cause significant capacity-demand imbalances. Adverse weather conditions particularly cause temporary and substantial reductions in airspace and airport capacity. Other factors that contribute to ATFM challenges are the number of runways available, ATC capacity, airspace restrictions and restrictions as to which aircraft can follow an aircraft of a given class are factors that determines the number of flights that can depart from or arrive at a certain airport and the number of aircraft's that traverses a particular sector of the airspace at a particular period of time (Bertsimas and Patterson, 1998; Odoni, 1987; Vossen et al., 2012).

In as much as ATFM has specific objectives and challenges to be addressed, it remains very necessary that responsibilities relating to decision making are shared between a number of stakeholders. Collaborative Decision-Making (CDM) is of paramount importance if appropriate ATFM actions are to be implemented. The main idea behind CDM is that through the exchange of data and collaboration among stakeholders, air traffic flow managers would be able to come up with better and effective decisions. The CDM objectives include *generating information based on the data from airspace system provided by airspace users; distributing back the same information to both users and traffic managers, in order to create common situational awareness, tools and procedures that allow for direct response to congestion by airspace users; and finally collaborating with air traffic flow managers in formulating ATFM actions* (Vossen et al., 2012). Hoffman et al. (1999) pointed out that the paradigm shift from central planning to collaborative flow management is a better way to alleviate congestion problems as it allows airlines to have more control, flexibility and input in the decision-making process through their control centers.

The problem of managing air traffic in order to maintain a safe and efficient flow of aircraft throughout the airspace is referred to as the Air Traffic Flow Management Problem

(ATFMP). Odoni (1987) came up with three approaches for solving ATFMP categorised as long, medium and short-term approaches. Long-term approaches (usually 5 - 10 years) try to solve the ATFMP by increasing capacity through construction of new airports and additional runways, improvement of traffic control techniques and the introduction of new technologies. Medium-term approaches (6months - 2years) can be seen as economic and administrative measures taken to solve the ATFMP. This can be done either by modifying the traffic flow patterns via temporary redistribution; flight diversion to off-peak times; or regulation of the rate of departure and arrival in order to mitigate the congestion problems. Short-term approaches (implemented daily within a time frame of at most 6 - 12 hours) are those actions that are efficiently applied to control the air traffic flow. The goal is to *match the demand with available capacity over time and across the various components of the ATC and airports network* (Vossen et al., 2012). These actions mitigate congestion problems that arise as a result of weather disturbances or unpredictable disruptions in the shortest possible times. In fact, most of the optimisation models discussed in Chapter 3 addressing ATFMP focus more on the short-term approach. Precisely, Bertsimas and Patterson (1998) presented a generalised model for the ATFMP wherein they illustrated how the model can be slightly changed to accommodate several options of the basic problem, remarkably, the proposal of two approaches namely, sector and path approaches, for incorporating rerouting option into the basic model especially when there is bad weather condition (Agustín et al., 2009; Bertsimas and Patterson, 1998; Odoni, 1987; Vossen et al., 2012).

Based on the 2004 report of the EUROCONTROL Performance Review Commission, the problem of the en route capacity constraints may take at least another decade to be solved because of its persistent nature. As a result of the occurrence of both airport and en route airspace constraints at the same time, ensuring a safe and efficient flow of aircraft through the airspace becomes a great challenge and a much more complicated task for traffic managers especially when trying to resolve the capacity-demand imbalances. One of the major challenges air traffic managers usually encounter is the problem of finding optimal scheduling strategies that mitigate the impact of congestion problems on ATC system. Hence, the need to come up with good and optimal scheduling ATFM strategies that do not only alleviate congestion problems but also minimise delay costs while satisfying the airport and en route airspace capacity constraints (Agustín et al., 2009; Bertsimas et al., 2011)

1.2 Motivation for the Study

Air traffic congestion is currently a major problem facing the world's air transportation systems, especially in most developed countries. Resultantly, Air Traffic Flow Management has become a necessity because of its central role in mitigating congestion problems as well as alleviating the associated costs. It is no surprise that it has become a fertile ground for research work, which has attracted widespread interest on the ATFM problem in both theory and practice. Given the recent trend in the aviation sector and the fast-growing demand for air transportation, it becomes very needful to come up with strategies that optimise the available resources to match the current air traffic demand as these will not only reduce the impact of congestion problems on ATC systems but will also meet up with the future demand for air transportation as well as the projected growth of the airline industry. It is quite unfortunate that most of the systems and infrastructure of the air transportation industries in most developed and developing countries do not have the capacity to meet the rising demand for air transportation as a result of the inefficiency of the flow management system. (Bertsimas and Odoni, 1997; Ronald, 2010).

To the very best of the researcher's knowledge, most of the models developed so far for the ATFMP considered instances from American and European airports and airspace where ATFMP is assumed to be of great importance. Contrarily, the reality on the ground as revealed in some reports is that most airports and airspace within other developing countries are experiencing similar problems and these need to be addressed urgently to meet with the future demand. Before the existing air traffic management system can be upgraded, an in-depth understanding of the basic ATFMP, as well as the existing optimisation models, needs to be developed. This will allow for a critical analysis of the problems that a given country is experiencing and then try to either extend or adapt it to address the challenge. This study is thus motivated by the afore-identified gaps.

1.3 Problem Statement

Air Traffic Flow Management is a planning activity designed specifically to address capacity-demand imbalances, which occur either when capacity is reduced or when demand is high (Bertsimas and Patterson, 1998; Odoni, 1987). The primary aim is to protect the Air Traffic Control (ATC) system from overloading as well as to mitigate congestion problems by limiting the resulting delays.

Until the late nineties, ATFM focused mainly on *airports congestion* and the most popular approach, by far, was the *allocation of ground delays (holding) to departing flights* (Odoni, 1987). Apart from the ground holding approach, they also used other available control options which include: *speed control of airborne aircraft; metering of air traffic*[2]*; en route airborne holding especially, near or inside terminal airspace and rerouting* to resolve congestion problems. In recent times, research findings and current practices have shown that congestion problems also affect en route air sectors as a result of overcrowded airspace, that is, capacity reduction in the airspace.

[2] controlling the rate at which aircraft go past a given point in airspace

Bertsimas et al. (2011) pointed out that *traffic congestion at these sectors is as critical an issue as congestion in terminal airspace around major airports*. Thus, airport and airspace capacity constraints have been listed as the major causes of congestion problems (Agustín et al., 2009; Bertsimas and Odoni, 1997; Odoni, 1987).

One of the ways to alleviate these problems on ATC system as well as maintaining operational and economic efficiency is by *optimising operations for given demand and capacity levels through optimal control of the air traffic flow in order to meet up with the demand across the various components of the ATC and airports network over time using available resources* (Odoni, 1987). However, system disruptions and other unforeseen factors that cause substantial reductions in airspace and airport capacity create unpredictable situations that require robust solution method. There are however other factors that determine the number of flights that can depart from or arrive at a certain airport as well as the number of aircraft that can are allowed in a particular sector of the airspace at a particular period of time. These include *availability of runways, ATC capacity, airspace constraints, limitations on aircraft of the same class as to which aircraft can follow another* (Ganu, 2008; Odoni, 1987; Vossen et al., 2012).

Given the presence of airport and en route airspace constraints, it becomes very difficult for the traffic managers to adequately manage the traffic flow efficiently. This exposes a need to find or develop optimisation techniques (approaches) as well as optimal scheduling ATFM strategies that address these problems while satisfying the en route airspace and airport capacity constraints which are also capable of handling large scale problems taking into consideration the level of uncertainty with some of these problems.

The basic ATFMP can be stated as follows:

Given an airspace system, consisting of a set of airports, airways, and sectors, each with its own capacity for each time period, t, over a time horizon of T periods, and given a schedule of flights through the airspace system during T, we want to assign ground and airborne delays to the flights in a way that satisfies all the capacity constraints while minimising a function of the cost of the total delay assigned

Thus, the problems to be addressed are stated as follows:

i) Conducting a critical survey of existing optimization models and solution techniques that address the ATFMP.

ii) Finding a better strategy for reducing air traffic delay and congestion problems in en route airspace by improving on existing strategies and optimization models.

iii) Application of the sector and path-based approaches proposed by Bertsimas and Patterson (1998) in developing new integer programming formulation for the ATFMP.

iv) Incorporating additional variables and constraints into the proposed IP formulation for the ATFMP in order to address other aspects of the basic problem.

1.4 Objectives of the Study

The main objective was to study the ATFMP as well as existing models and solution approaches that have been devised to solve the problem. More specifically the study sought:-

1. To review the existing optimisation models for solving ATFMP based on the type of the problem they address and solution methodology. The primary aim of the review is to provide insight into the problem formulation, solution methodology and identify the best available model that will serve as underlying model when focusing on solving the Air Traffic Flow Management Problem within the African region using instances from any African airports and airspace.

2. To improve on already existing models by extending the model to allow for other control options and modeling variations with focus on the deterministic case before extending to probabilistic environment.

3. To investigate the computational performance of the existing optimisation models in addressing the ATFMP on different data instances and draw valuable conclusions from the computational results. More precisely, to see how efficient and realistic it is to obtain integral solutions especially when solving both small and large scale instances.

1.5 Significance of the Study

Ronald (2010) noted that one of the major significance of studying ATFMP is to *empower the decision-making process of air traffic flow management by filling the knowledge gap and emphasising the need for integration in the ATFM decision-making process.* Thus, the research findings as well as the valuable conclusions deduced from the computational results of the existing ATFMP models presented in this study are of paramount importance to the air transportation industry as it aids in the ATFM decision making.

Although, most of the aviation industries are using a computerised procedure, software and other tools to address the ATFMP, a closer look into the current practices in the Air Traffic Control Systems reveals that the different optimisation models reviewed in this study are well suited to be the underlying principles, that is the optimisation engine, for the ATC system. This is evident from the fact that most of the models were validated using real-world and artificially constructed (generated) data sets. Thus, the existing models and optimisation solution approaches of the ATFM will help in improving the efficiency of the current and future traffic flow management practices if much attention is given to them. Moreover, the ideas and underlying principles employed in the problem formulation of the existing optimisation models for the ATFMP is not restricted only to the air transportation industry but can also be used in other areas like manufacturing and road transportation industries where goods and services are flowing through a dynamical system with different types of capacitated elements (Bertsimas and Patterson, 1998).

1.6 Scope of the Study

ATFMP is complex and fraught with different subproblems that need to be addressed. The scope of the study is therefore limited to the aspect of air traffic management that focuses more on managing the air traffic flow while satisfying airport and en route airspace capacity constraints. The study does not analyse the technicality involved in the data exchange of the ATC and ATFM system; rather it gives an insight into the current ATFM practice and problem within the context of South Africa. Also, the study does not cover the analysis of the various traffic flow management models, but it focuses only on the optimisation models for solving the ATFMP based on the different categories of problems under consideration. Although the review of the optimisation models, covers both deterministic, stochastic, static, dynamic and heuristics approaches for solving ATFMP, the study only considers the deterministic case for the proposed formulation of the ATFMP that includes rerouting and other control options. Moreover, the study does not provide a mathematical analysis of the mathematical models given in this thesis, rather an insight is given on the concepts of polyhedral combinatorics and integer programming that aptly capture the ATFMP to help in understanding the modeling and optimisation of the ATFMP and proposed integer programming formulation.

The researcher also encountered challenges with accessing air traffic managers within his proximity in order to get useful information. Moreover, access to real-life data or already generated data instances that are referenced in other academic works was almost impossible given that the confidentiality level vis a vis such data is very high. Other than that, obtaining access and license approval for the stand-alone modeling systems that make use of CPLEX optimisation solver, which was mostly used in literature, is a lengthy process given my status as an amateur in the usage of CPLEX optimiser and associated modeling systems. Thus, most of the computational results are obtained using the students stand-alone solver, which has a restriction on the number of variables and constraints that it can handle when using the students' stand alone version. This does not allow for large instances of data.

1.7 Research Contribution

It was the fervent hope of the researcher to contribute the following to ongoing research in this area:

1. The critical analysis of the existing approaches, methodologies and optimisation techniques that have been (and continue to be) used to address the ATFMP presented herein in this thesis not only provides a detailed overview of the modeling and optimisation of the ATFMP but also provides further directions and recommendations for future research especially with applications to developing countries.

2. New integer programming formulations were proposed for reducing the air traffic delays and congestion problems based on two different approaches by extending the basic air traffic flow model formulation to account for rerouting and other control options.

3. The proposed deterministic integer programming formulation is envisaged to be as strong as the underlying model as most of the inequalities (constraints) are facets defining for the convex hull of solutions. Thus, the proposed formulation when tested on real life instances can be used for effective decision making when managing the flow of traffic in the air.

4. Small data instances depicting a typical ATFM problem were artificially constructed based on the reviewed concepts in order to investigate the computational performance of the basic ATFMP model (both for the MIP and LP relaxation case) in obtaining integral solutions.

1.8 Thesis Outline

The remaining chapters of this thesis are organized as follows:

Chapter 2 describes the basic concepts in modeling and optimisation, optimisation process and techniques, integer programming and polyhedral combinatorics. Thereby, giving the background knowledge that is needed for proper understanding the problem under consideration.

Chapter 3 provides an overview of the ATFMP by giving a detailed description of the problem as well as the initiatives that have been put in place to address the ATFMP from both a theoretical and practical point of view. The different existing approaches, methodologies and optimisation techniques that have been used to address the problem were critically reviewed.

Chapter 4 provides an insight into the modeling approach for the proposed sector and path-based approaches and the mathematical formulation of the new integer programming deterministic model for the reducing air traffic delay and congestion problem. In Chapter 5, the computational results obtained from the artificially constructed datasets for validation purposes were presented and discussed.

Lastly, Chapter 6 contains the summary of the thesis, concluding remarks and recommendations for future research.

1.9 Chapter Summary

This chapter was an introduction into the thesis, wherein background information to the study was provided. An overview of the Air Traffic Management was presented with the goal of distinguishing between the two components, ATC and the ATFM. The chapter clearly stressed on the projected growth in the air traffic and demand for air transportation which poses a great threat of severe congestion to the ATC system as well as the current challenges encountered by traffic managers in finding optimal scheduling strategies that mitigate the impact of these congestion problems on the ATC system.

The chapter also outlined the specific objectives and challenges of ATFMP and presented an overview of the approaches that have been put in place to address the problem. The motivation, objectives, significance and scope of the study were also outlined in this chapter.

Chapter 2

Conceptual Framework

This chapter is an overview of the basic concepts, approaches and terminologies that have been and continue to be employed in order to address the ATFMP. In Section 2.1, the concept of modeling and optimisation, modeling and optimisation process as well as techniques are briefly discussed. Key concepts in Integer Programming and Polyhedral Combinatorics, which is the underlying principle for most of the ATFMP optimisation models, are defined in Section 2.2. Most of the definitions, theorems, description and illustration given in this chapter closely follow the ones presented in Odoni (1987); Sarker and Newton (2007); Wolsey and Nemhauser (1988).

2.1 Modeling and Optimisation Techniques

2.1.1 Introduction to the Optimisation Problem

Several problems of interests that need to be solved (addressed) abound in every day life. A unique class of problem that has attracted interest of researchers, problem-solving practitioners and policy makers over the last decade are those problems whose solutions yield minimum cost or maximum profits. These problems are generally referred to as optimisation problems. Optimisation problems can be found in all spheres of life and disciplines of study including biological sciences, engineering, management science, information and communication technology, computer science, humanities and social science etc due to the fact that there is always the need for planning and decision making (Ali et al., 2015). The basic idea of optimisation is finding the best possible solution to a given problem by careful examination of all possible solutions and show that the selected solution is indeed an optimal (best) solution (Ali et al., 2015; Sarker and Newton, 2007).

Researchers and problem-solving practitioners in most cases use mathematical models to represent the problem of interest before attempting to solve it. The complexity of real-world problems makes it difficult to develop mathematical models that can address all aspects of the problem and in most cases an impossible one. However, problem-solving practitioners and researchers tend to formulate simpler versions of the problem by making several approximations and assumptions of which the solutions to the reduced problem may likely differ to that of the original problem (Sarker and Newton, 2007).

2.1.1 Definition. *A model is an abstraction of a problem of interest. It is to explain, predict, or control the behavior of the entity modelled and forms an essential part of the* *process of solving that problem optimally (Chinneck, 2006; Sarker and Newton, 2007).*

Three major types of models exist namely; analog, iconic, and mathematical or symbolic models but the one that is of interest to optimisation is the mathematical model as it uses a *set of functional relationships and mathematical symbols to represent physical quantities and allows for manipulation of the object that is been modeled.* Mathematical models or any type of models developed for solving problems must be *simple, robust, adaptive, complete and user-friendly* (Chinneck, 2006; Sarker and Newton, 2007).

2.1.2 Definition. *A mathematical model is a mathematical representation of the real system or object of the problem and is able to present the important features of the system in a form that is easy to interpret (Sarker and Newton, 2007). In other words, it is an orderly and structured manner in which real-life problems might be tackled.*

Mathematical models are generally classified into four categories: descriptive models, optimisation or normative models, often referred to as the prescriptive model, heuristic models and predictive models. Descriptive models allow for representation of physical situations in a visual mode. Optimisation models seek to optimise, that is either to 'maximise' or to 'minimise', an objective (mathematical) function of a decision variables, with respect to some constraints. Heuristic models use intuitive rules which are guided by common sense to look for solutions to a problem. In most cases, the solutions of optimisation models are optimal while the solutions of heuristics model are said to be near-optimal. Predictive models are developed in such a way that decisions about future trends can be made (Sarker and Newton, 2007; Luke, 2009).

The focus of this study lies mainly on *optimisation and optimisation models.* There are three major components to an optimisation (mathematical) model namely *objective function, decision variables and set of constraints to be satisfied* (Sarker and Newton, 2007). The objective function of any mathematical model can either be a function of a one or multiple variables. In any case, the optimisation problem is referred to as single and multi-objective optimisation problems respectively. The decision variables used in the mathematical model can also vary depending on the nature of the problem under consideration. That is, the variables can be of the following forms: real, integer or a combination of both. Moreover, the optimisation problem can either be constrained or unconstrained. For the constrained optimisation problems, the left hand side of the constraint function in the constraint parts of a the model is usually separated with any of the inequalities: ($\leq$), ($\geq$), and ($=$). A typical (standard) mathematical model is of the form:

$$\text{Maximse} \quad f(x) \tag{2.1.1}$$

subject to

$$h_i(x) \leq H_i, \quad i = 1, \cdots, m \tag{2.1.2}$$

$$g_j(x) = G_j, \quad j = 1, \cdots, n \tag{2.1.3}$$

$$x \geq 0, \quad x \in \mathbb{R}^n \tag{2.1.4}$$

The objective function, (2.1.1), is a function of *a single variable, x*. The objective can also be to minimise $f(x)$ where minimisation is simply a negation of maximisation, that is, min $f(x) = -\max(-f(x))$. The constraints, (2.1.2) - (2.1.4), comprises of *constraint functions, h_i and g_j, which are general functions of the variable (otherwise expressed as an unknown, decision variable or sometimes as a parameter); known constants for deterministic problems, H_i and G_j; and a non-negativity constraint, $x \geq 0$,* which is necessary for most real-world problems since most variables cannot be negative due to the fact that the problems tends to either minimise costs or maximise profits which cannot be assumed to be negative. This standard model is subject to change in any of the following form: it can *contain variable bounds instead of a non-negativity constraint or any other constraint or contains only the standard model with or without* (2.1.2) *and* (2.1.3). The last case is applicable only to multiple variables (Sarker and Newton, 2007).

Supposing that x^* denotes set of variables say $x^* = \{x_1, x_2, \cdots, x_p\}$, Then the standard mathematical model for multiple variables will be of the form:

$$\text{Maximse} \quad f(x^*) \tag{2.1.5}$$

subject to

$$h_i(x^*) \leq H_i, \quad i = 1, \cdots, m \tag{2.1.6}$$

$$g_j(x^*) = G_j, \quad j = 1, \cdots, n \tag{2.1.7}$$

$$x^* \geq 0 \tag{2.1.8}$$

Mathematical models are generally characterised by the input sets, parameters, objective function, decision variables and constraints. A detailed description of these characteristics as well as general assumptions relating to mathematical model formulation can be found in (Sarker and Newton, 2007, pg 6-7).

2.1.2 Classification of Optimisation Problem

General optimisation problems are classified based on the following: the number of objective functions; constraints and function type, that is, the nature of expressions for the objective and constraint functions; nature of the problem under consideration; and nature of the decision variables (Adewumi, 2010; Chinneck, 2006). More precisely, the problem classification as summarised in Fig 2.1 is categorised as follows: objective classification, problem or constraint classification, variable classification and function classification

1. **Objective Classification:** As alluded in the previous section, optimisation prob-

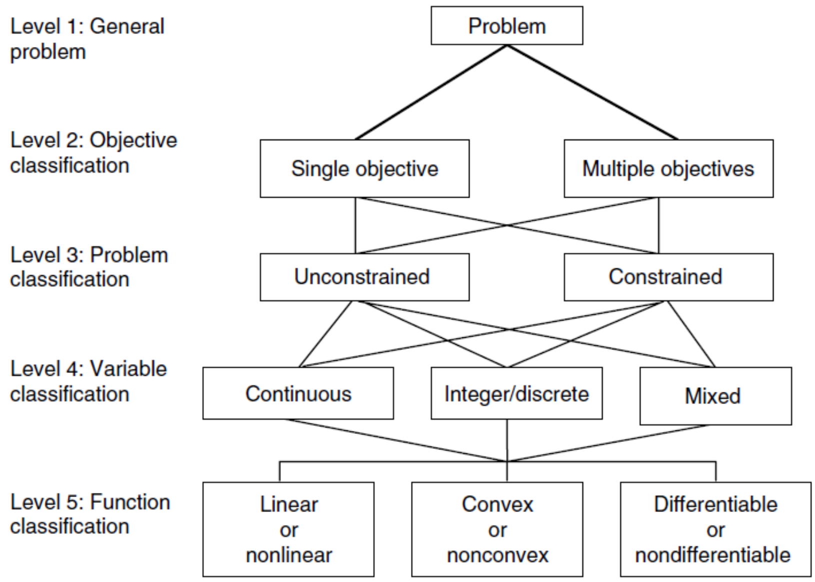


Figure 2.1: Classification of Optimisation Problem (Adewumi, 2010; Sarker and Newton, 2007).

lems can be categorised as either single or multiple objective problem depending on whether the function is of single or multiple variable. Moreover, the problem can be classified based on the objective type which can either be *to maximise or minimise.*

2. **Problem or Constraint Classification:** The problem classification is an indicator that shows whether the problem is constrained or unconstrained that is whether it contains constraints or not. Although most real world problems are believed to be constrained, the study of unconstrained optimisation problems is also of paramount importance to researchers as most of the unconstrained optimisation techniques provide solution procedures for constrained problems. In constrained optimisation problems, the constraints can either be hard or soft. Hard constraints are *those constraints that the conditions must be satisfied in the final solutions* (Sarker and Newton, 2007). In other words, hard constraints defines the feasibility of the solutions to be obtained and cannot be breached while soft constraints are *those constraints that can be violated under certain conditions or with a certain penalty* (Sarker and Newton, 2007).

3. **Variable Classification:** As has been mentioned in the preceding section, an optimisation problem can only have real, integer or mixed (combination of real and integer) variables. These variable classifications are generally referred to as continuous, integer or discrete, and mixed variables respectively. Optimisation problems with continuous variable are referred to as *Real-valued Programming Problems* (Chinneck, 2006; Sarker and Newton, 2007). The nature of the problem implies that the solution set must emanate from the set of real numbers. The optimisation problems with *integer or discrete variables is also known as combinatorial problem or inte-*

ger programming problem. The solution space consists of an object from a finite or infinite set-typically an integer, permutation, or graph (Sarker and Newton, 2007). Most practical real-world problems fall under combinatorial problems since it involves permutation, combination (or even mixture of both) of some set of items or even both subject to certain constraints. In particular, ATFMP is an example of combinatorial problem since it involves scheduling, assignment and planning.

4. **Function Classification:** The function classification is based on the nature of the objective and constraint functions which can either be *linear, nonlinear or both* as well as the mathematical properties of the functions which are considered important from the solution approach point of view (Chinneck, 2006; Sarker and Newton, 2007). Given any optimisation (mathematical) model and based on the function classification, the model is said to be a linear programming model if all the functions are linear. On the other hand, it is referred to as a nonlinear model if one or more of the functions of a model has some elements of nonlinearity. However, several special cases have been recorded where an optimisation model has linear constraints but with either a quadratic or geometric objective function. The optimisation problems that arises as a result of this classification are referred to as *Linear Programming Problem, Nonlinear Programming Problem, Quadratic Programming Problem and Geometric Programming Problem respectively* (Sarker and Newton, 2007).

   Since most optimisation techniques are developed based on the assumption that the function is convex, convexity has been considered as an important property in classical optimisation. Thus, optimisation problems can also be classified as either convex or nonconvex optimisation problem depending on the convexity properties of functions. Moreover, the problems can also be classified as non-differentiable or differentiable based on the solution approaches which can either be derivative free (without derivatives) or with derivatives (Chinneck, 2006; Sarker and Newton, 2007).

   In addition to the the above classification, the following function properties: unimodal, multimodal, static and dynamic are also considered when classifying optimisation problems. Unimodal functions are functions with *only one optimum (peak) solution* while multimodal functions are those functions with '*more than one optimum solution*. The solution can either be a local or global optima. Static functions are functions that does not change with time while dynamic functions changes over time (Sarker and Newton, 2007).

Deterministic Optimisation versus Stochastic Optimisation Problems

The distinguishing factor between deterministic and stochastic optimisation problems is whether the associated data for the given problem (capacities of the system) are known or uncertain. For deterministic optimisation problems, the data for the given problem is known accurately while in stochastic optimisation or optimisation under uncertainty, the capacities of the system are assumed to be probabilistic and the uncertainty is incorporated into the model. However, for most real-world problems, the data is not always deterministic due to a number of reasons. Czyzyk et al. (1997) noted that in most real world problems, the data cannot be known accurately due to *simple measurement or prediction error coupled with the fact that some of the data contains information about the future which is uncertain* (Czyzyk et al., 1997). For the case of stochastic optimisation, uncertainty is incorporated into the model formulation using the concept of scenario trees with associated weights. Techniques like robust optimisation (estimation) can be applied to such problems when the parameters are known up to certain limits with the aim of finding feasible optimal solution for all the data instances. The goal of stochastic programming as opined by Czyzyk et al. (1997) is "*to find some policy that is feasible for all (or almost all) the possible data instances and optimises the expected performance of the model by taking advantage of the fact that the probability distributions governing the data are known or can be estimated*" (Czyzyk et al., 1997; Chinneck, 2006; Sarker and Newton, 2007).

2.1.3 Optimisation Process

An important stage that is crucial to many institutions, organisation and production companies is the optimisation phase or process. This is because of the central role that it plays during the design and decision making processes. Markedly, it helps the decision makers in coming up with realistic solutions to most complex management problems that are of utmost concern to them. Although there are several ways to address the problems that are of utmost concern to decision makers, the best way one might ever think of may not be necessarily unique or obvious. Thus, optimisation provides a step by step process of finding the best way (optimal solution) to the complex real-world problems (Adewumi, 2010; Chinneck, 2006; Sarker and Newton, 2007).

Until now, problems of utmost concern are addressed using either a qualitative approach (where problem solvers rely on previous experience and personal judgements for decision-making when handling similar situations) or a quantitative approach, which involves the use of already developed techniques to solve the problems when they are either recurrent or complex involving many variables.

Sarker and Newton (2007) is of the opinion that quantitative approaches are more appropriate and efficient when addressing complex problems than qualitative approaches since *it provides a better structured and logical path through the decision-making process.* The authors also pointed out that the process of decision-making begins once a real-world phenomenon or problem that is of interest is identified (Sarker and Newton, 2007). This process can also be referred to as the mathematical modelling process. The identified problem is extensively studied and analysed with the aim of gathering necessary information that is needed for proper understanding of the problem as well as accessing available information relating to the problem which will be of help when formulating the mathematical model. Thereafter, an optimal solution has to be found using the model by following *a sequence of mathematical evaluations, performing several iterations and testing in order to validate the formulated model* (Sarker and Newton, 2007).

Subsequently, the model assumptions are examined at various stages using datasets that are not necessarily meant for the study and those that have been collected specifically for the problem at hand to ascertain whether the assumptions provide valid answers to the problem under consideration (Chinneck, 2006). These stages are often referred to as dry and wet run. Sensitivity analysis is then carried by the decision makers with the aim of identifying the decision variables that have the most significant impact on the solution and investigating how the solutions are affected by slight changes in the input data (Sarker and Newton, 2007). This analysis actually helps in ascertaining the *robustness of the preferred option as well as determining how sensitive the choice of the option is to slight changes in the data and assumptions* (Chinneck, 2006). The solution is then tested against past and future behaviour as well as observations of actual performance over a period of time in other to validate the robustness of the model before final implementation (Chinneck, 2006; Sarker and Newton, 2007).

The decision-making process as shown in Fig 2.3 involves six stages or phases namely, *Identifying and clarifying the problem; Defining the problem; Formulating and constructing a mathematical model; Obtaining a solution to the model, Testing the model, evaluating the solution, and carrying out sensitivity analysis; Implementing and maintaining the solution* (Chinneck, 2006; Sarker and Newton, 2007). None of the steps can be said to be the most demanding item in the process but in the case that the problem is not well defined, the analysis will eventually lead to wrong results. Hence, extra care is required when defining and analysing the problem. Figures 2.2 and 2.3 illustrates the modelling and optimisation process but the core optimisation process are steps 2-5 of Fig 2.3.

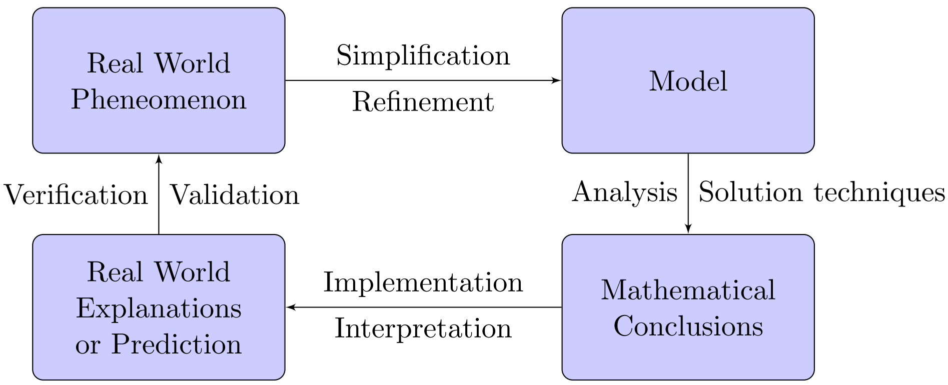


Figure 2.2: Illustration of Modelling Process.

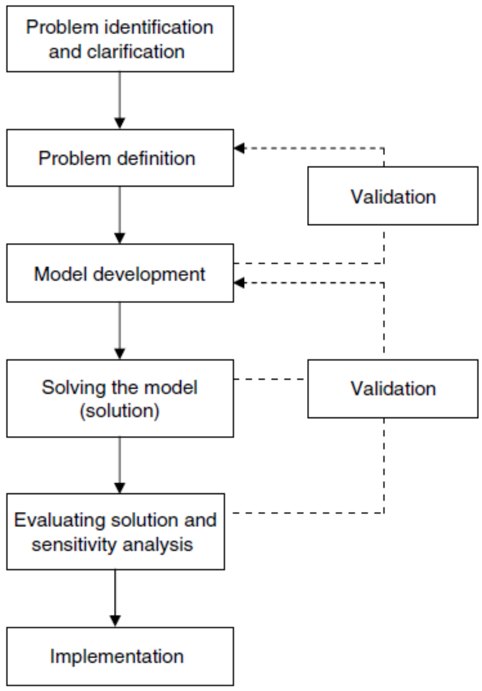


Figure 2.3: Decision-Making Process (Sarker and Newton, 2007).

2.1.4 Optimisation and Modeling Techniques

There are several ways to address the identified complex problems. However as previously highlighted, the best strategies may not necessarily unique or obvious. The process of finding this best way or optimal solution to identified problems is known as optimisation and the choice of suitable optimisation techniques or methods depends on the type of optimisation problem under consideration.

Optimisation techniques are applied to solve most complex real-life and industrial problems (Ali et al., 2015). The existence of optimisation techniques and mathematical models for solving real-world problems can be traced back to several decades ago when concepts of calculus of variation and differential calculus was the basic tool used for finding maximum or minimum values of functions when solving theoretical problems as well as those arising from practical situations (Sarker and Newton, 2007). For instance, the famous secretary problem was solved by Johannes Kepler in 1613 and as far back as 300 BC Euclid showed that *the minimal distance between two points is the length of straight line joining the two* and years later in 1657, Kepler proved that *light travels between two points through the path with shortest length* (Shodhganga, 2010; Sarker and Newton, 2007). Subsequently, in the early nineties, people like H. L. Gantt, F. W. Harris, A. K. Erlang amongst others derived mathematical equations for solving most of the problems that were of utmost concern. For instance, Gantt charts developed in 1900 are used efficiently today to schedule jobs on machines while F. W. Harris, on the other hand derived, *the well-known economic order quantity in inventory management, used today in determining the most economic quantity of an item to order from a vendor.* A. K. Erlang's formulation is used to analyse *problems encountered by callers to an automated telephone switchboard which in turn gave rise to the recent queuing/waiting line analysis* (Sarker and Newton, 2007; Chinneck, 2006).

Several other methods were later introduced for solving both constrained and unconstrained optimisation problems. L. Kantorovich in 1939 and George B. Dantzig in 1947 developed *an algorithm, known today as simplex algorithm, for solving linear programming problems* (Sarker and Newton, 2007). J. L. Lagrange introduced the method of *Lagrange multipliers* for solving constrained optimisation problems while L. A. Cauchy in 1847 proposed *an iterative method for solving systems of equations* (Chinneck, 2006). The proposed method is known today as the *gradient method* and has been applied to solve unconstrained optimisation problems. Interior point optimisation methods, which are based on the polynomial time algorithm developed by N. Karmarkar in 1984 are also part of the solution methods for most optimisation problems (Chinneck, 2006; Sarker and Newton, 2007).

Other advanced techniques have also been developed over the past decades for solving optimisation problems, amongst these, *a dynamic programming algorithm* developed by Richard Bellman in 1957. These techniques provide procedural steps that help in exploring a solution search space in order to obtain a feasible solution that optimises given objective functions subject to certain constraints. Moreover, the recent development of *modern heuristic techniques such simulated annealing (SA), tabu search (TS), genetic algorithms (GA), neural computing, fuzzy logic, particle swarm and ant colony optimisation (PSO and ACO)* has helped problem-solving practitioners and researchers in addressing more complex optimisation problems like blood assignment in blood banking system and space allocation problem and many others (Adewumi, 2010; Adewumi and Ali, 2010; Olusanya et al., 2015).

These optimisation techniques are classified into two major categories: Classical or exact optimisation techniques and heuristics (meta-heuristics) techniques which are outlined below. Exact optimisation techniques are briefly discussed while heuristics methods are outlined with little or no description on the different techniques. More details on the different methods can be accessed in any optimisation and modeling textbook.

2.1.4.1 Classical or Exact Optimisation Techniques

Exact optimisation techniques are suitable when finding optimal solutions to both single and multiple objective optimisation problems. These techniques obtained optimal solutions to given problems via an iterative process starting from an initial solution. The different methods are briefly described below.

- **Direct methods:** These are brute force approaches used in exploiting the nature of functions which do not require evaluation of derivatives. The solution approaches used depend on the type of the problem, that is multiple or single objective optimisation problem. For single optimisation problems, the golden-search, a general-purpose single-search technique, or quadratic interpolation method can be used while univariate search or random search methods are used for multi-objective optimisation problems (Chinneck, 2006).

  The golden-search method starts with upper and lower bound points (two initial guess), followed by selection of the interior point which is chosen according to the golden ratio given by $\frac{\sqrt{5}-1}{2}$ and the function is evaluated at new points and accordingly the either the lower or upper bound is changed. The premise of quadratic or three points interpolation method is that there will be only one quadratic connecting three points and a quadratic polynomial which often provides a better approximation to the objective function near optimal solution (Shodhganga, 2010).

  The random search method is used for multiple objective optimisation problems where the solution space are searched repeatedly by evaluating the function at randomly selected values of independent variables with the goal of finding an optimal solution. It works well for non-differentiable and discontinuous functions since it does not require the gradient of the problem to be optimised but the only setback is that the behaviours of functions at already evaluated points are not accounted for by the method. On the other hand, the univariate search method optimises the objective function with respect to a single variable at a time. The multivariable problem are reduced to series of single-variable optimisation problems using this search method and as the variables are interchanged, the process converges to an optimal point. (Chinneck, 2006; Luke, 2009; Shodhganga, 2010).

- **Gradient methods:** Gradient methods, as stated earlier are *iterative methods for solving systems of equations* (Shodhganga, 2010). These methods obtain optimal solutions to real-valued optimisation problems of the form $\min\limits_{x \in \mathbb{R}^n} f(x)$ based on the knowledge of derivative information with the search directions defined by the function's gradient at the current state. The gradient descent and the conjugate gradient methods are some examples of the gradient method (Shodhganga, 2010; Weisstein, 2015).

- **Linear programming methods:** There are specific methods that apply to Linear Programming Problems and this includes the graphical method, simplex method (algorithm) and many more. The graphical solution method is used mostly for two or three dimensional linear programming problems where the feasible solution region and optimal solution are determined by *plotting the graph of the equality and inequality constraints and by finding the point in the feasible region for which the objective function is optimal respectively* (Chinneck, 2006; Shodhganga, 2010). This method is not applied regularly in practice although it is one of the effective methods used for solving LP problems.

Simplex Method (Algorithm): As revealed in the introduction to this section, simplex algorithm was invented by George Dantzig in 1947 to solve LP problems. The algorithm determines the feasible solution set of any given LP problem by evaluating the solutions gathered from the corner points with the aim of finding the optimal point from the solution space. The simplex method tests the adjacent corner points of the feasible solution set (usually a polytope[1]) in sequential order in such a way that any new corner point, the objective function is either the same or improved. To do this, the $n-m$ variables are set to zero in order to obtain the basic solution for m linear equations with n unknowns, the m equations for m remaining unknowns are finally solved until the optimum is found. The *zero variables* are called the *nonbasic variables*, while the *remaining m variables* are referred to as the *basic variables* (Shodhganga, 2010). The basic solution including the optimum is obtained when all the basic variables are non-negative (Sarker and Newton, 2007; Shodhganga, 2010).

Before applying simplex method to linear programming problems, it is necessary to ensure that the LP problem is converted into the standard form where the objective is maximised, the constraints are inequalities and the variables are all nonnegative by adding an external (slack or surplus) variable that converts the inequality constraints into equality constraints as well as applying transformation to those variables with unrestricted signs in the formulation. This method provides optimal solution for linear programming problems in an extremely efficient manner and very efficient in practice unlike the graphical method which has very limited practical utility (Chinneck, 2006; Shodhganga, 2010)

Interior point methods: Unlike the simplex method that solves only linear programming problems, the interior point methods are used to solve both linear and nonlinear constrained optimisation problems. As far back as 1960, the interior method was solely used for solving nonlinear constrained optimisation problems but after the invention of fast interior methods for solving LP problems in 1984 by Narenda Karmarkar, interior methods are now applied to almost all kind of optimisation problems (Sarker and Newton, 2007; Shodhganga, 2010).

The logic behind the interior method is that the best solution to any given optimisation problem are obtained through searching the interior of the feasible region. The interior methods that give solutions to linear programming problems are the *Khatchian's ellipsoid method and Karmarkar's projective scaling method* (Shodhganga, 2010). Primal-dual method is also another interior point method. The interior method has a lot of advantages over the simplex method both in terms of complexity and the fact that it is not affected by the problem of degeneracy. This is

[1]See Remark 2.2.5 of Section 2.2

evident from the fact that the estimated run time needed to solve a LP problem of size n by the simplex method is of the order $2n$ while that of the interior point method is of the order of $ni : i = 2$ or 3. In essence, the complexity of interior methods is polynomial for both average and worst cases and that of simplex method is polynomial for average case and exponential for worst case since *polynomial time algorithms are computationally superior to exponential algorithms for large linear programming problem* (Chinneck, 2006; Sarker and Newton, 2007; Shodhganga, 2010).

- **Branch and Bound ($B\&B$) Algorithm :** This method is amongst several others that are used for solving integer programming (combinatorial) and real-valued problems. The $B\&B$ algorithm was proposed in 1960 by A. H. Land. The goal of the method is to find a value x that optimises the objective function within the feasible region. The algorithm consists of *a systematic enumeration of individual solutions by means of state space search.* This is based on the fact that the outlined integer solutions has a tree-like structure with the set of individual solutions forming the rooted tree with the full set at the root (Sarker and Newton, 2007). Thus, the tree-like structure is assumed to have a root node, leaf node and bud node with parent node closest to the root node and child node closest to the leaf node as well as edges representing the algorithm's partial or complete solutions (Chinneck, 2006; Sarker and Newton, 2007).

  The core idea is to to grow the trees in stages with only the auspicious nodes grown at each stage and avoid the entire tree as much as possible. The most auspicious nodes are determined by putting a restriction, that is assuming a bound, on the best objective function value that can be obtained if the nodes are grown to the later stages. Chinneck (2006) noted that the $B\&B$ algorithm *explores the branches of this tree, that is the subsets of the solution set and before the candidate solutions of a branch are enumerated, the branch is first checked against upper and lower estimated bounds on the optimal solution* (Chinneck, 2006; Shodhganga, 2010). The solution is discarded if it cannot produce a better solution than the best one found thus far by the algorithm. Thus, the two important aspects of the algorithm are *Branching and Pruning.* Others include *Bounding and Fathoming, a by-product of the bounding function calculations* (Chinneck, 2006). In summary, the algorithm performs *a recursive search through the tree-like structure from top to down,* formed by the branching after which the other aspect of the algorithm follows suit (Chinneck, 2006; Sarker and Newton, 2007; Shodhganga, 2010). The key terms that are associated with the branch and bound algorithm are briefly defined below.

  - ***Branching:*** The splitting of the solution search space into smaller spaces
  - ***Pruning:*** The cutting off and permanently discarding of nodes (individual solutions) that that the algorithm can show that it will never be feasible or optimal by keeping track on the bounds (Chinneck, 2006; Sarker and Newton,

2007).

– ***Bounding:*** This is *when the bound on the best value attained by a growing node is estimated* (Chinneck, 2006) .

– ***Fathoming:*** The is case where *the expansion of a bud node can be halted* when the potential objective function value obtainable by expansion can be seen directly.

– ***Bounding Function:*** This is the method of *estimating the best value of the objective function obtainable by giving a bud node further* (Chinneck, 2006; Sarker and Newton, 2007)

- **Dynamic Programming** is mostly assumed to be a general concept rather than a strict algorithm although often times in literature it is referred to as the dynamic programming algorithm. The concept is mainly applied to problems with a temporal structure with the aim of splitting the problem into disparate sub problems before it is finally solved. This method of solving the problem is distinct in the sense that the sub problem for the last period is solved. The optimal solution for the last period is obtained by first determining the optimal solution for the last but one period. The sequence continues like that until all the sub problems are finally solved (Chinneck, 2006; Shodhganga, 2010).

2.1.4.2 Heuristics and Meta-heuristics Optimisation Techniques

Heuristic and Metaheuristic optimisation techniques or algorithms serve as an alternative for the classical optimisation techniques discussed in the previous section. The need for heuristic techniques particularly arises when classical techniques are either too slow or cannot solve large instances of a given problem. Precisely, when classical techniques fails to find exact solution to a given optimisation problem. Hence, heuristic techniques are developed for solving complex optimisation problems much quicker or finding alternative solutions to the given problem where an exact solution does not exist, that is, they search for the best feasible solutions to the given optimisation problem where the complexities involved does not permit an optimal solution (Adewumi, 2010). This is done by either a trial and error method or trade off on optimality, completeness, accuracy, or precision for speed and execution time. Heuristics techniques are often referred to as a short cut method. Although the techniques are used to address most real-world complex problems and have emerged as one of the strongest producers of good or near-optimal solutions to the problem, most of the heuristics are exposed to several pitfalls, which however go beyond the scope of this study. Some pitfalls have strong underlying theory while others are just based on some rules of thumb learned empirically without the influence of theory. Examples of such techniques include greedy algorithm, alpha-beta pruning and all other search algorithms (Adewumi, 2010; Luke, 2009).

On the other hand, metaheuristics is an advancement of heuristic algorithms. The phrase *meta* simply means *higher level* (Luke, 2009), thus, meta-heuristics simply means higher-level heuristics, implying that meta-heuristics algorithms can perform better with less computational efforts than simple heuristics algorithms, classical optimisation techniques, and iterative methods when looking for feasible solutions to a given problem. Hence, they are very useful for solving most real-world problems as well as combinatorial optimisation problems. The method of implementation in most cases takes the form of stochastic optimisation in the sense that the solutions depends on the set of random variables generated (Luke, 2009). Essentially therefore, metaheuristic algorithms are approximate, non-deterministic and not problem specific.

Metaheuristics as defined by most authorities is a process that "*guides subordinates heuristics by bringing together different concepts and ideas for the purpose of investigating and exploiting the search space in order to obtain efficient near optimal solutions*" (Beheshti and Shamsuddin, 2013; Laporte and Osman, 1995). It endeavours to maintain a balance between exploitation and exploration of the local neighbourhood structures of the solution space . Exploration searches for the global optimum solution point while exploitation seeks to find more '*promising*' local neighbourhood as most superior solutions can be found within such neighbourhood (Adewumi, 2010; Luke, 2009).

Metaheuristics algorithms are classified based on certain properties they possess or exhibit. Some of the techniques include *Genetic Algorithm(GA), Simulated Annealing(SA), Tabu Search, Ant Colony Optimisation(ACO), Particle Swarm Optimisation (PSO)* among others that fall into different classification based on their properties. The classification is outlined below along with examples of metaheuristics optimisation techniques for each classification (Adewumi, 2010; Luke, 2009).

- **Local search-based versus Global search-based Metaheuristics**

  Metaheuristics are characterised based on the search strategies they employ in finding an optimal solution to given optimisation problem.

  Local search algorithms are used when there are too many candidate solutions to a given optimisation problem. The idea is to *move from solution to solution in the search space, that is the space of candidate solutions, by applying local changes, until an optimal solution is found or time bound is elapsed* (Luke, 2009). On the other hand, global search algorithms are metaheuristics that improve local search heuristics in order to find better solutions. A typical example of local search-based metaheuristics is the *Hill Climbing (HC) Method* while *Simulated Annealing (SA), Tabu Search (TS), Iterated Local Search, Variable Neighborhood Search, and Greedy Randomised Adaptive Search Procedure (GRASP)* can be classified as both local or global search-based metaheuristics. Moreover, global search-based metaheuristics that are not local search-based are generally known as population-based metaheuristics. See population-based metaheuristics below (Adewumi, 2010; Adewumi and Ali, 2010; Luke, 2009).

- **Single-solution versus Population-based Metaheuristics**

  Single-solution approaches focus more on *improving and adjusting a single candidate solution in the search space while Population- based search methods, which are inspired by the social behaviours or natural concepts, seek to maintain and improve multiple candidate solutions* (Luke, 2009). The search techniques are mostly based on the population characteristics. In particular, single-solution based metaheuristics are local search-based heuristics. Examples of single-solution based metaheuristics include *SA, Iterated Local Search algorithms, Variable Neighbourhood Search and Guided Local search while GA, Evolutionary Computation, PSO, ACO, Artificial Bee Colony, Social Cognitive Optimisation on the other hand are examples of population-based metaheuristics* (Beheshti and Shamsuddin, 2013; Luke, 2009). PSO, ACO, Artificial Bee Colony, Social Cognitive Optimisation are classified under Swarm Intelligence Metaheruristics since it is a *collective behaviour of decentralised, self-organised agents in a population or swarm* (Adewumi, 2010; Luke, 2009; Beheshti and Shamsuddin, 2013).

Metaheuristics can be combined with other optimisation techniques so as to improve on the search mechanism. This idea of *combining metaheuristics with algorithms from mathematical programming, constraint programming, and machine learning is referred to as hybridisation and the resulting algorithm or metaheuristic is often called a hybrid metaheuristic* (Luke, 2009). On the other hand, memetic algorithms are those algorithms that represent the synergy of evolutionary or any population-based approach with separate individual learning or local improvement procedures for problem search (Adewumi, 2010; Luke, 2009)

2.2 Integer Programming and Polyhedral Combinatorics

In this section, key concepts in Integer Programming (IP) and Polyhedral Combinatorics are briefly discussed. The basic concepts and definitions and theorems closely follow the one presented in (Wolsey and Nemhauser, 1988).

In the previous section, IP problems were defined as those optimisation problems with integer variables and whose solution space consists of *an object from a finite or infinite set, say integer, permutation or graph* (Sarker and Newton, 2007). In fact, an IP problem is one that seeks to optimise, that is *to minimise or maximise an objective function*, say $f(x)$, subject to the following conditions:

- The vector x must satisfy the given set of constraints functions, that is, x must satisfy the system of linear inequalities ($Ax \geq b$) (Sarker and Newton, 2007).

- x must be a non-negative ($x \geq 0$)
- x must be an integer.

If the last condition is omitted, then the resulting problem is said to be a LP problem and the resulting formulation is said to be relaxed (LP relaxation of the IP).

Consequently, the link between integer programming and polyhedral combinatorics will now be discussed starting with definition of basic concepts.

2.2.1 Definition. *A set of points of dimension m, that is, $x_1, \cdots, x_m \in \mathbb{R}^n$ is said to be affinely independent if the unique solution to the equation below is $\lambda_i = 0,\ \forall\ i = 1, \cdots, m$.*

$$\sum_{i=1}^{m} \lambda_i x_i = 0, \sum_{i=1}^{m} \lambda_i = 0 \tag{2.2.1}$$

2.2.2 Remark. *For linear independence, the second equation is not included. Affine independence is used because IP deals mostly with inequalities.*

2.2.3 Definition. *The 'maximum number of linearly independent row (columns) of A' is the rank of A, denoted by rank(A) while the 'maximum number of affinely independent points in $\mathbb{R}^n$' is $n + 1$, that is n linearly independent points and the zero vector (Wolsey and Nemhauser, 1988).*

2.2.4 Definition. *Let P be a set, P is said to be a* ***polyhedron*** *if $P \subseteq \mathbb{R}^n$ is a set of points satisfying a finite number of linear inequalities, that is, $P = \{x \in \mathbb{R}^n | Ax \leq b\}$, with (A, b), an $m \times (n + 1)$ matrix (Ganu, 2008; Wolsey and Nemhauser, 1988).*

2.2.5 Remark. *A polyhedron is a 'convex set'. Moreso, it can be bounded. A polyhedron is bounded if there is an $\alpha \in \mathbb{R}^1_+ : P \subseteq \{x \in \mathbb{R}^n : -\alpha \leq x_j \leq \alpha$ for $j = 1, \cdots, n\}$. A bounded polyhedron is called a* ***polytope*** *(Ganu, 2008; Wolsey and Nemhauser, 1988).*

2.2.6 Definition. *A polyhedron, P, is of* ***dimension*** *k, denoted dim(P) $= k$, if the maximum number of affinely independent points in P is $k + 1$ (Wolsey and Nemhauser, 1988). In other words, the dimension of P is just one less than the cardinality of the maximal set of affinely independent points in P (Ganu, 2008). $P \subseteq \mathbb{R}^n$ is said to be full dimensional if dim(P) $= n$.*

2.2.7 Proposition. *If $P \subseteq \mathbb{R}^n$, then dim(P)+rank$(A^=, b^=) = n$. $(A^=, b^=)$ is the equality set consisting of the rows corresponding to $a^i x = b_i\ \forall\ x \in P$ (Wolsey and Nemhauser, 1988).*

Proof. See Proposition 2.4 of Wolsey and Nemhauser (1988) for proof. □

2.2.8 Definition. *The inequality, $\pi x \leq \pi_0$ or (π, π_0) is a **valid inequality** for P if it is satisfied by all points in P. In other words, (π, π_0) is a valid inequality if and only if* $\max\{\pi x : x \in P\} \leq \pi_0$ *(Wolsey and Nemhauser, 1988).*

2.2.9 Definition. *If (π, π_0) is a valid inequality for P and $F = \{x \in P : \pi x = \pi_0\}$, then F is called a **face** of P.*

2.2.10 Remark. *The following remark is as a result of the above definitions.*

- *The inequality (π, π_0) represents F. F is said to be proper if $F \neq \emptyset$ and $F \neq P$.*
- *Moreover, F represented by (π, π_0) is nonempty if and only if* $\max\{\pi x : x \in P\} = \pi_0$. *Hence, (π, π_0) supports F.*
- *A face of P is said to be a **facet** if dim(F) = dim (P) - 1.*
- *P can described by the facets of P.*

2.2.11 Definition. *Given a set $S \subseteq \mathbb{R}^n$, a point $x \in \mathbb{R}^n$ is said to be a **convex combination** of points of S if there exists a finite set of points $\{x_i\}_{i=1}^t$ in S and a $\lambda \in \mathbb{R}_+^i$ with* $\sum_{i=1}^t \lambda_i = 1$ *and* $x = \sum_{i=1}^t \lambda_i x_i$ *(Wolsey and Nemhauser, 1988).*

2.2.12 Definition. *The **convex hull** of S, denoted by conv(S), is the set of all points that are convex combinations of points in S (Wolsey and Nemhauser, 1988). In other words, conv$(S) = \{y \in \mathbb{R}^n : y = \sum_{i=1}^t \lambda_i x_i\}$ where $x_i \in S$ and $\lambda_i \in \mathbb{R}_+$ $\forall i$ with $\sum_{i=1}^t \lambda_i = 1$.*

If S is defined as $S = P \cap \mathbb{Z}^n$, then convex hull of S, *conv*(S) , corresponds to the convex hull of integral polyhedron on $\mathbb{R}^n$. See Figure 2.4.

2.2.13 Definition. *$x \in P$ is said to be an **extreme point** of P if $\nexists\, x_1, x_2 \in P$, $x_1 \neq x_2$ such that $x = \frac{1}{2}x_1 + \frac{1}{2}x_2$.*

2.2.14 Remark. *If $\pi x \leq \pi_0$ is valid for S, then it is also valid for conv(S). Furthermore, if $\pi_x \leq \pi_0$ defines a face of dimension $k-1$ of conv(S), then there exists k affinely independent points $x_1, \cdots, x_k \in S$ such that $\pi x_i = \pi_0$ for $i = 1, \cdots, k$ (Wolsey and Nemhauser, 1988). More details on this can be seen in the proof of Propositions 6.5 and 6.6 of (Wolsey and Nemhauser, 1988, pg 107-108).*

2.2.15 Remark. *Two valid inequalities, $\pi x \leq \pi_0$ and $\gamma x \leq \gamma_0$ are said to be equivalent if $(\gamma, \gamma_0) = \lambda(\pi, \pi_0)$ for some $\lambda > 0$, otherwise they are not equivalent. If they are not equivalent and there exists $\mu > 0$ such that $\gamma \geq \mu\pi$ and $\gamma_0 \leq \mu\pi_0$, then $\{x \in \mathbb{R}_+^n : \gamma x \leq \gamma_0\} \subset \{x \in \mathbb{R}_+^n : \pi x \leq \pi_0\}$. In this case, we say that $\gamma x \leq \gamma_0$ dominates $\pi x \leq \pi_0$ or $\pi x \leq \pi_0$ is dominated by $\gamma x \leq \gamma_0$ (Wolsey and Nemhauser, 1988).*

2.2.16 Definition. *A maximal valid inequality for S is one that is not dominated by any other valid inequality (Wolsey and Nemhauser, 1988).*

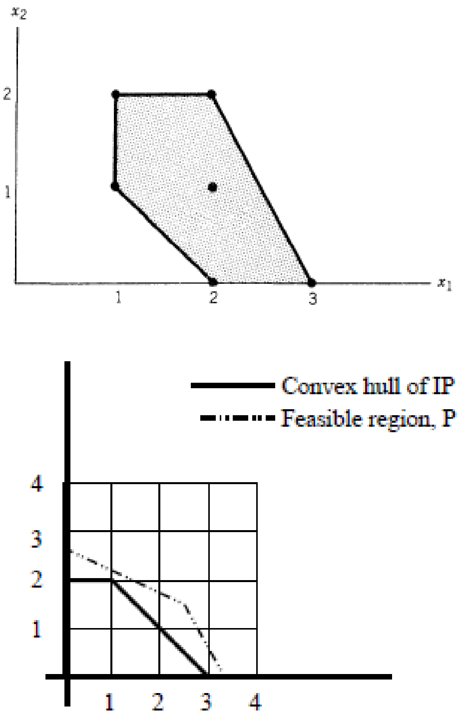


Figure 2.4: Convex hull of integral solutions in $\mathbb{R}^n$ (Ganu, 2008; Wolsey and Nemhauser, 1988)

2.2.17 Remark. *Any maximal valid inequality for S defines a nonempty face of conv(S), and the set of maximal valid inequalities contains all of the facet-defining inequalities for conv(S) (Ganu, 2008; Wolsey and Nemhauser, 1988).*

2.2.18 Theorem. *Given $S = P \cap \mathbb{Z}^n \neq \emptyset$, $P = \{x \in \mathbb{R}^n_+ : Ax \leq b\}$ and any $c \in \mathbb{R}^n$. It follows that:*

- *The objective function value of the IP problem is unbounded from above if and only if the objective of the LP relaxation is unbounded from above.*
- *If the LP relaxation has a bounded optimal value, then it has an optimal solution (namely, an extreme point of conv(S)), that is an optimal solution to the IP.*
- *If x^* is an optimal solution to the IP problem, then x^* is an optimal solution to the LP relaxation (Wolsey and Nemhauser, 1988).*

In other words, if $P = \{x \in \mathbb{R}^n_+ : Ax \leq b\}$ is the feasible set of the LP relaxation of the IP problem max$\{cx : cx \in conv(S)\}$ and the LP on the convex hull of S has an optimal solution, x^* (namely an extreme point of conv(S)). Then, x^* will be an optimal solution to the IP on S (Ganu, 2008; Wolsey and Nemhauser, 1988).

Proof. See Theorem 6.3. of Wolsey and Nemhauser (1988) for the proof. □

The above theorem simply indicates that we can solve the IP problem by solving the LP relaxation. Thus, the solution to an IP problem can be obtained directly from its LP relaxation.

2.2.19 Remark. *In a situation where each of the corner points of the set of feasible points to the LP relaxation is integral, then there will always be an integral optimal solution,* x^**, to the LP relaxation. Since every integer point feasible to the IP problem is contained in the feasible set of points of the LP relaxation, then the optimal solution for LP relaxation,* x^* *must be the optimal integer solution for the IP problem (Wolsey and Nemhauser, 1988).*

This merely is a cursory explanation on the theorem, more details can be located in Chapters 1 and 2 of Wolsey and Nemhauser (1988). The basic concepts illustrated in this section form the underlying principle for the best identified optimisation model for the ATFMP that contributed positively to the computational performance of the model.

Based on the basic concepts described above, the best option or technique in IP is to *find a formulation (possibly a strong formulation) for which the feasible set of points to its LP relaxation fits tightly around the convex hull of integer solutions* (Hoffman, 1997). This closeness of fit is measured by the *value gap*[2] while the strongest formulations are *ones that define facets of the convex hull of integer solution* (Hoffman, 1997; Wolsey and Nemhauser, 1988).

IP problems are usually more difficult to solve than LP problems. Resultantly therefore, most of the algorithms for solving IP or MIP problems rely on branch and bound strategy by enumerating the set of feasible points and exploring the tree solutions in which each branch represents a restriction on the set of feasible points. Research findings shows that much can be done to simplify an integer programming problem after formulation before it is finally solved. A preprocessing phase involves all the steps or measures taken in order to simplify an integer programming problem before it is finally solved. It is an integral part of most optimisation solvers especially CPLEX. These measures include one of the following: fixing some of the variables a priori or even eliminating the variables altogether; examining the bounds and constraints for potential tightening and elimination respectively (Hoffman, 1997; Hoffman et al., 1999). All the techniques of integer programming as well as the concepts briefly discussed above are mostly applied when formulating the air traffic flow management problem.

[2] Value gap is the difference between the objective function values of the IP and LP relaxation. It is used for determining a stronger model when given two different problem formulation. A lower value gap indicates a stronger model (Hoffman, 1997; Wolsey and Nemhauser, 1988).

2.3 Chapter Summary

This chapter focused mainly on modeling and optimisation concepts. In summary, Figure 2.5 shows the classification of optimisation problems taxonomy with particular emphasis on the sub fields of deterministic optimisation with a mono-objective function.

The concept of optimisation problem, the three main components of mathematical model, the objective function, decision variables and a set of constraints to be satisfied, as well as the general characteristics and assumptions were introduced. The different optimisation problems classified by Sarker and Newton (2007) and Chinneck (2006) were briefly discussed.

The chapter also illustrated the various stages involved in optimisation since it is pivotal to many institutions and decision makers. The optimisation process helps in the decision making and design process by providing realistic solutions to most complex real-world optimisation problems. Different modeling and optimisation techniques that are classified under two broad headings; exact and heuristic techniques, were also discussed. Basic concepts in integer programming and polyhedral combinatorics were also defined and briefly discussed with the aim of illustrating the relationship between integer programming and polyhedral combinatorics.

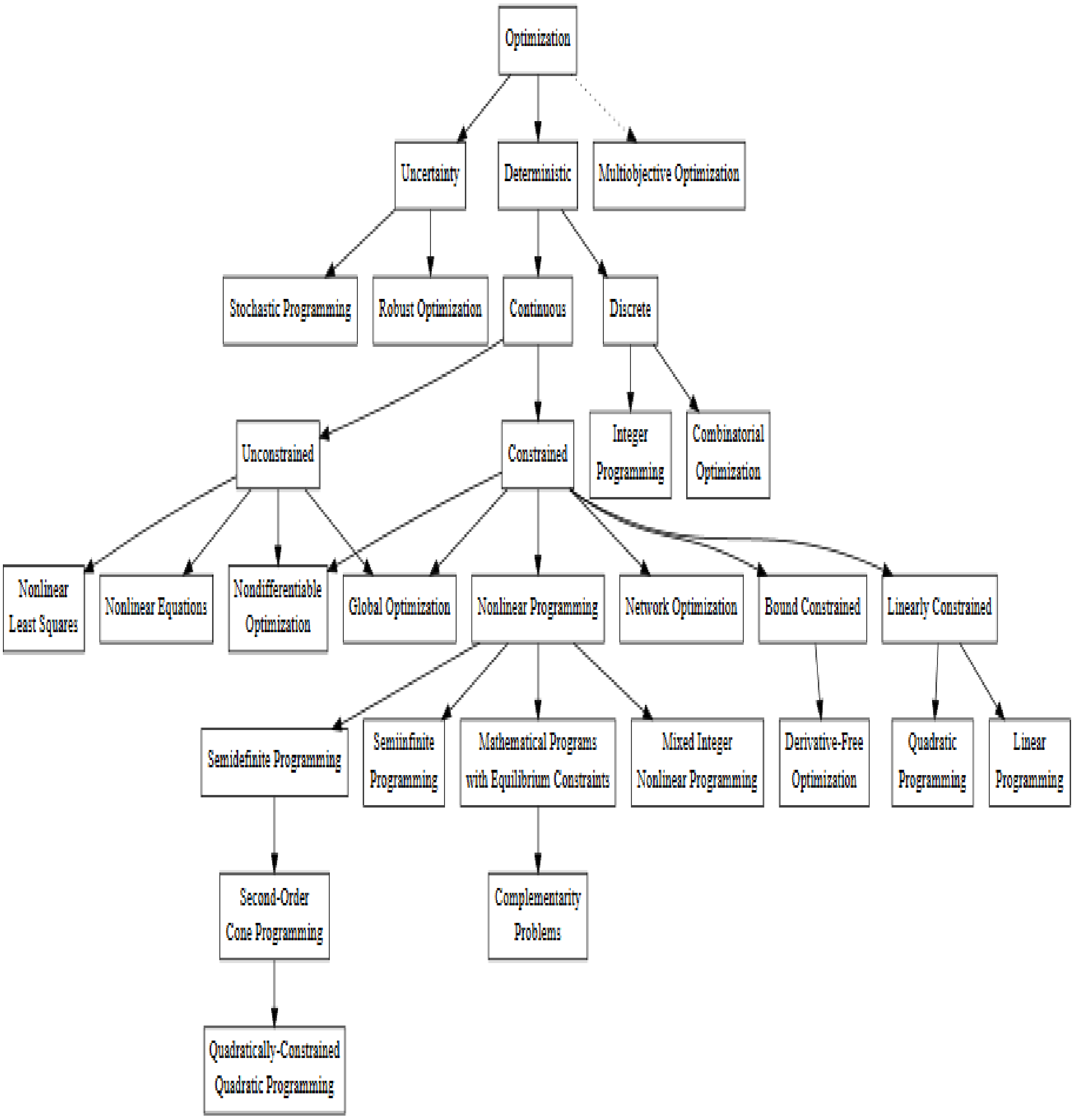


Figure 2.5: Taxonomy of optimisation problems (Czyzyk et al., 1997).

Chapter 3

Methodologies for the Air Traffic Flow Management Problem

This chapter is an exploration of existing approaches, initiatives and methodologies that have been used to address the ATFMP in the context of South Africa from both a theoretical and practical point of view. A review of the optimisation models taxonomy for solving the ATFMP is also presented in Section 3.3.

3.1 Air Traffic Flow Management Problem Description

The ATFMP can be described in terms of a network flow model or better still from a graph theoretical point of view considering the geographical representation of an airspace which can be represented in terms of graphs and network flows (Delahaye and Puechmorel, 2013; Helme, 1992). See Figures 3.1 , 3.2 and 3.3 for illustration.

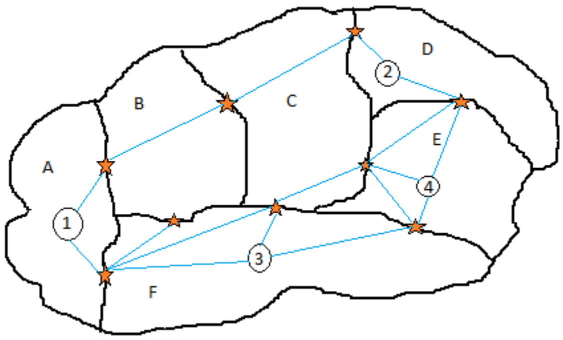


Figure 3.1: Illustrative Example: A geographical representation of an airspace.

In order to describe ATFMP as a network problem, the elements of the network as visualised in Fig 3.3 are describe first for proper understanding. They include airports (A/P), airways (A/W), sectors (S/R) and waypoints (W/P).

1. Airports: Aircrafts make use of airport for departure and arrival (to destination), thus an airport is an origin for some flights and destination for others. Moreover, aircrafts fly through waypoints.

2. Waypoints: A waypoint is a *trans-shipment node through which traffic flows. It may*

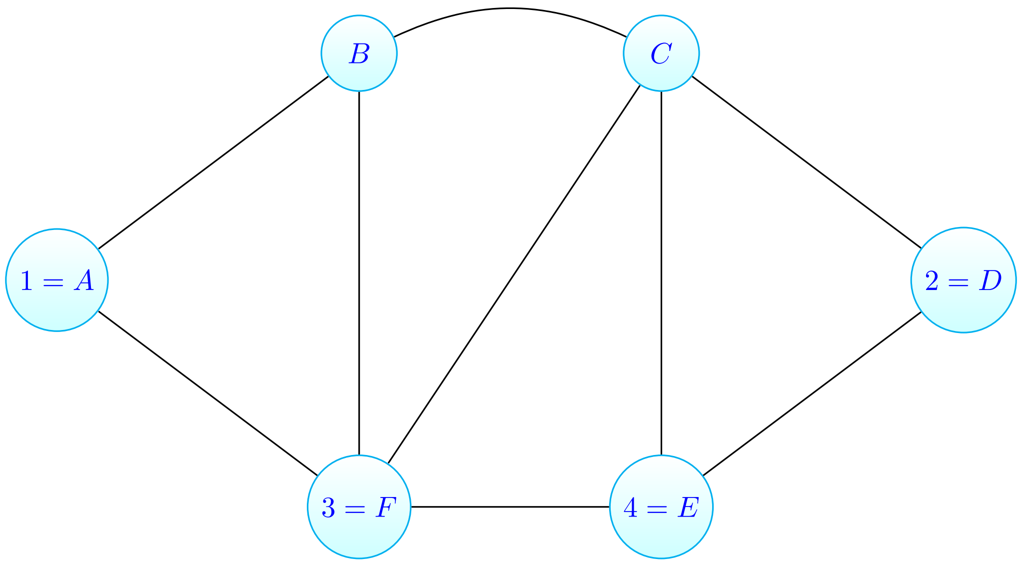


Figure 3.2: A graph representation of the geographical airspace

be a departure fix, an arrival fix, a point of intersecting airways, or any point in the airspace that one wishes to designate (Helme, 1992). It is neither an origin nor a destination for any flight.

3. Airways: An airway is a recognised or predefined route followed by aircraft while traversing the airspace to its destination.

4. Sectors: Sectors are *collections of waypoints and adjacent sections of airways* (Helme, 1992).

3.1.1 Remark. *A node in the network can represent an airport or a waypoint while a link or arc represents an airway between two nodes.*

3.1.2 Remark. *Airports are the sources and sinks of flows on the network while airways represents the arcs on which flows travel. On the other hand, the nodes of the network are the waypoints. The airways either merge, intersect or diverge at such points Odoni (1987).*

The most critical of these elements are the airplane terminals and sectors since they constitute the chief bottleneck of the system (Odoni, 1987). This is a consequence of limitations on the quantity of aircraft that can leave or touch base at a specific air terminal and additionally the quantity of flying machine that is permitted to be in a specific area at a specific time frame. In this manner, every air terminal has a landing and a flights limit (See Figure 3.3 where R and D are the servers connected with Airport 1) which shifts with time contingent upon various elements such as poor climate conditions; runway setup being used; ATC partition prerequisites and techniques; sorts of aircraft and portion of every kind of the movement; runway geometry; and human components (execution of ATC controllers and pilots) that change after some time (Odoni, 1987). Besides, deciding airplane terminals limits is regularly described by instability since one can't foresee precisely the climate conditions at any given point in time despite the fact that now and again, the limits are known ahead of time (Bertsimas and Patterson, 1998; Bertsimas and Odoni, 1997; Odoni, 1987; Vossen et al., 2012).

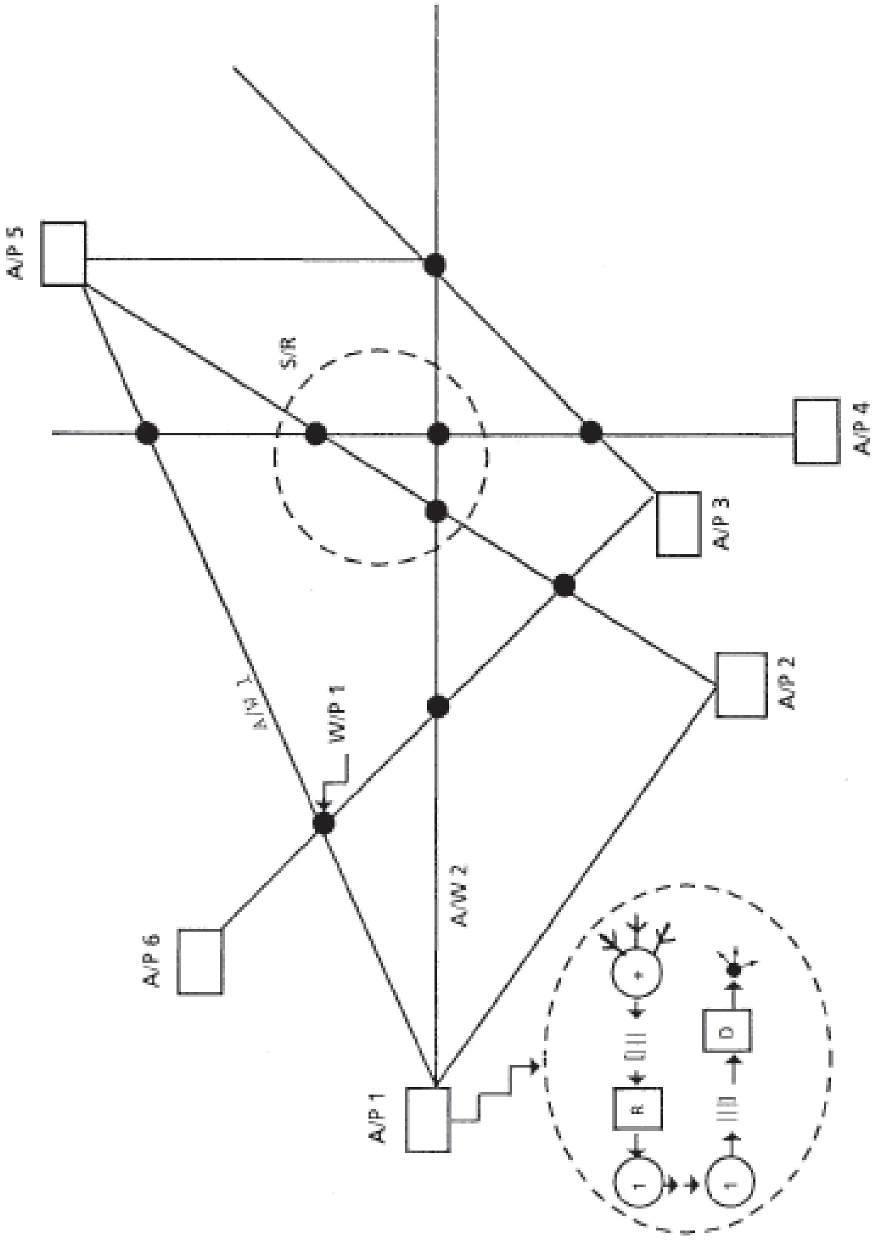


Figure 3.3: Network representation of ATC system for ATFMP purposes (Odoni, 1987).

On the other hand, aviation routes, waypoints and sectors limits are controlled by the greatest number of aircraft that can navigate these components at a specific time frame period. On account of sectors, it is controlled by the maximum number of aircraft that can be in a segment for a time period (Helme, 1992; Odoni, 1987). Odoni (1987) noted that the limits offered by these components practically speaking as a rule surpasses the movement streams that can be created by and consumed at the air terminals Odoni (1987). In any case, sectors what's more, waypoints situated at key positions and near bunches of air terminals individually are regularly barred.

As already noted in Section 1.1 ATFM tries to reduce delays that arise as a result of congestion problems which usually occur when the demand for the systems infrastructure exceeds capacity. Taking into account the presumption that the limit of airways is not generally the reason for congestion, Odoni (1987) points out that delays in the ATC framework can happen as a consequence of an unfavorable demand-to-capacity relationship at the accompanying ATC components:

(i) Departing airport (as a result of congestion at the departure runways)

(ii) En route air shafts (airways) on account of waypoint congestion.

(iii) En route airways as a result of congestion at the en route airspace sector

(iv) Arrival airport (due to congestion of the arrival runways) (Odoni, 1987).

Further, Odoni (1987) stresses that once there is no great activity stream administration framework, the effect of delays subsidiary with (i) - (iv) would be significantly felt in the congested zones, that is, on the runway, preceding take-off; on the aviation routes, through airborne holding close to a congested waypoint or before entering a congested division; and in terminal zones, through airborne holding and vectoring before arriving at the airplane terminal of destination (Odoni, 1987). In any case, a great stream administration framework with the open data concerning the system tries to moderate the effect of clog and delays and additionally implementing any of the control activities so as to minimize the related expenses (Bertsimas and Odoni, 1997; Odoni, 1987; Vossen et al., 2012).

1. **Ground holding or Gate Holding:** Delaying the departure times of flights by not permitting the aircrafts to leave its departure gate at the scheduled time even though they are ready for departure (Bertsimas and Odoni, 1997).

2. **Traffic Metering:** is the regulation of the rates of aircraft flow through particular focuses on the system, for example, flight runways or various transit waypoints (Odoni, 1987).

3. **En route re-routing:** This is the modification of routes for selected flights when it is noticed that the current route is no longer usable due to problems associated with poor weather conditions and other factors. This will enable the flights to boycott crowded en route areas.

4. **En route speed control restrictions:** imposing a restriction on the cruising speed of flights while en route so as to monitor their time of arrival at *a waypoint or terminal area of an airport* (Odoni, 1987).

5. **Air (High-elevation) holding and path-stretching maneouvers:** This is a means of holding a flight in the air at high altitude. Thus, delaying arrival to crowded areas as well as avoiding the cost associated with holding flights at low elevation.

   ATFMP, as already stated in Section 1.1, is the problem of managing the air traffic so as to ensure efficient and safe flow of aircraft throughout the airspace but in a more general sense, ATFMP is "*the problem of designing a flow management system that will minimise the cost of ATC delays described in (i) - (iv) within the shortest possible time subject to a set of operational and policy constraints*" (Odoni, 1987).

   The ATFMP can be addressed from a macroscopic or microscopic point of view at different stages depending on the type of control action to be implemented. The traffic flow management control actions to be implemented can be *pre-tactical, strategic and tactical in nature* where decisions of a more tactical nature require more detailed, microscopic models while those concerned with strategic flow management requires macroscopic models with a high level of aggregation (Vossen et al., 2012; Odoni, 1987).

   When considering the control actions described above, *actions (4) and (5) are of a tactical (fine-tuning) nature, (1) is of a strategic nature and (2) and (3) fall in-between* (Odoni, 1987). In Section 3.3, most of the models that were reviewed considered at least one of the control actions as possible flow management tools but most importantly, the models try to resolve the most important aspect of ATFM from a practical point of view, that is, the trade-off between ground-holding delays and airborne delays (Odoni, 1987). The trade off is important in that flights are held on the ground (when aircraft is stationery and its engines switched off) instead of the air, which is quite costly and somewhat unsafe. Trade off is thus crucial in that it saves costs to the airlines and the nation as well.

3.2 The Air Traffic Flow Management Initiatives

In the United States, Air Traffic Control System Command Center (ATCSCC), an arm of the Federal Aviation Administration (FAA), has the obligation of dealing with the air movement action inside of the National Airspace System (NAS). Their part is to adjust air traffic demand with the system limit in the NAS and are focused on dealing with the NAS in a sheltered, effective, and firm way with the point of minimizing delays and clog while augmenting the general utilization of the NAS (Vossen et al., 2012). Different units that assists in the air traffic control and administration incorporates the Airport Control

Towers, Air Route Traffic Control Centers (ARTCC), Terminal Radar Approach Control Facilities (TRACONs) amongst others (Vossen et al., 2012).

In South Africa, the Central Airspace Management Unit (CAMU) is accountable for air traffic flow and capacity management inside of the South African airspace in a joint effort with South Africa Air Traffic and Navigation Services (ATNS). ATNS is the sole business supplier of air activity, route and related administrations in charge of aviation authority in roughly 10% of the world's airspace (ATNS, 2011, 2015).

CAMU is essentially in charge of the airspace capacity management; slot allocation; adaptable utilization of airspace, and re-routing of activity influenced by unfriendly climate or confined airspace utilizing diverse fundamental and progressed ATFM systems (ATNS, 2011).

Ground Delay Programme was the first to be implemented by FAA & ATCSCC since the problem of airport congestion basically occur in USA via *the national ground holding policy using a computerised procedure in order to select appropriate ground-holds* (Vossen et al., 2012). To ensure safe and efficient traffic flow management practices in South Africa, CAMU already have in place different ATFM techniques for traffic management decision making which can either be strategic, pre-tactical or tactical flow management decision. These techniques include: *Ground Stops, Airspace Flow and Ground delay Programmes (which includes Ground Holding, Airborne Holding, Flight Cancellation, Rerouting, Swapping, Miles and Minutes-in-Trail (MIT & MINIT), Traffic Sequencing Programmes, Speed Control and other control actions) as well as ATFM tools like Airport Flow Tool (AFT), Thales's ATFM solution (EUROCAT and FLOWCAT), Airspace Management Tool (AMT), Metron Traffic Flow AND CAMU WEB* amongst others. See (ATNS, 2011, 2015) and Section 3.2.1 for more details on these techniques. Figure 3.4 illustrates the exchange of information through the different phases of the ATFM between the control managers, traffic managers, airline operators and other decision making bodies and stake holders.

3.2.1 Central Airspace Management Unit ATFM Techniques

The description of the CAMU ATFM advanced techniques outlined in this section follow closely the descriptions presented in ATNS (2011, 2015) and others found in literature. See ATNS (2011) for more details on the basic CAMU ATFM techniques (Miles-in trails, Minutes-in-trail, depature flow rates, sequencing traffic programmes)

1. **Ground Stops (GS)** are actualized at aerodromes[1] when major ATC blackouts or terrible climate conditions make limit diminishments in the system framework to

[1] Aerodrome is an area from which airplane flight operations occur, paying little or no attention to whether they include air cargo, travelers, or none of them (ATNS, 2011).

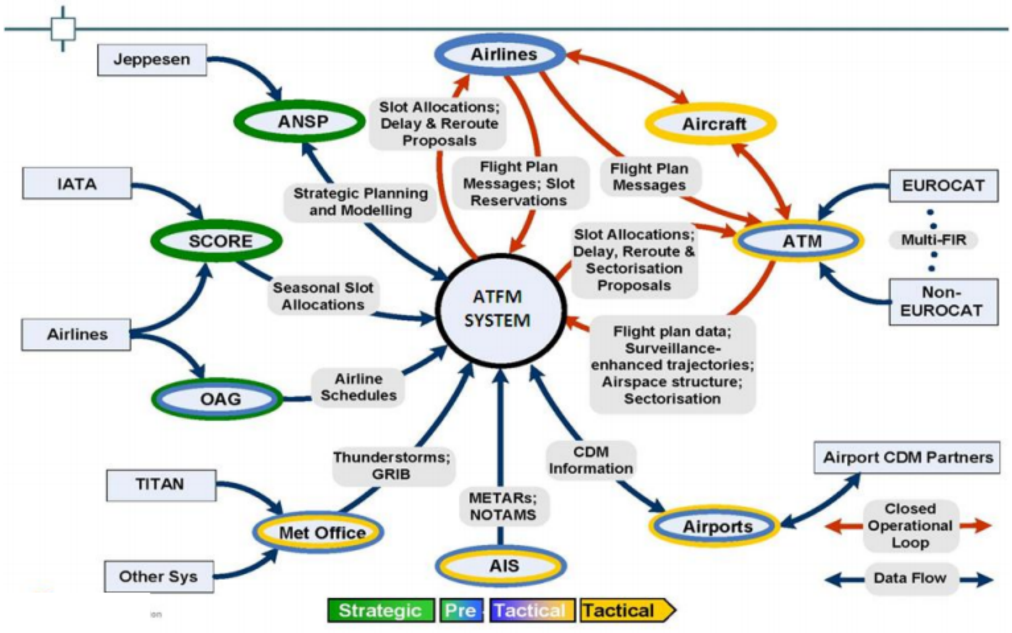


Figure 3.4: Screen shot of CAMU data exchange through the ATFM phases (ATNS, 2015).

the degree that traffic jam can happen at an aerodrome ATNS (2011).

2. **Ground Delay Programs (GDP)** are ground or entryway holding moves taken to delay the flights on the ground because of the limit imperatives at the landing or takeoff aerodromes. GDP's try to minimise costs by avoiding too much airborne delay or rerouting (ATNS, 2015).

3. **Airspace Flow Programs (AFP)** are those control activities actualized when there is a limit decrease in the airspace, that is, in the areas, waypoints or aviation routes. The influenced regions are distinguished as Flow Constrained Area (FCA)

4. **Airspace Management Tool (AMT)** is utilized predominantly for airspace management in order to ensure a safe and productive stream of flights. See Figure 3.5 for illustration. The AMT is composed in a manner that data in regards to the present air circumstance (concerning the flight arrangement tracks, graphical routes, maps, arrangements of flights, climate information, Storm TSA); the future air circumstance, and the sector/position/TSA activity burden, can be shown on the screen as an aide for traffic managers in executing flow management techniques. Different abilities include: snappy sorting/filtering, alerts, notices and stacking maps and also transmission of ATFM rerouting messages (ATNS, 2011, 2015)

5. **Airport Flow Tool (AFT)** is a pre-strategic propelled choice bolster device that resolves the capacity-demand imbalances within the South African airspace proficiently (see Figure 3.6) by observing the systems capacity and demand and implementing necessary traffic management initiatives. Airport Flow Tool

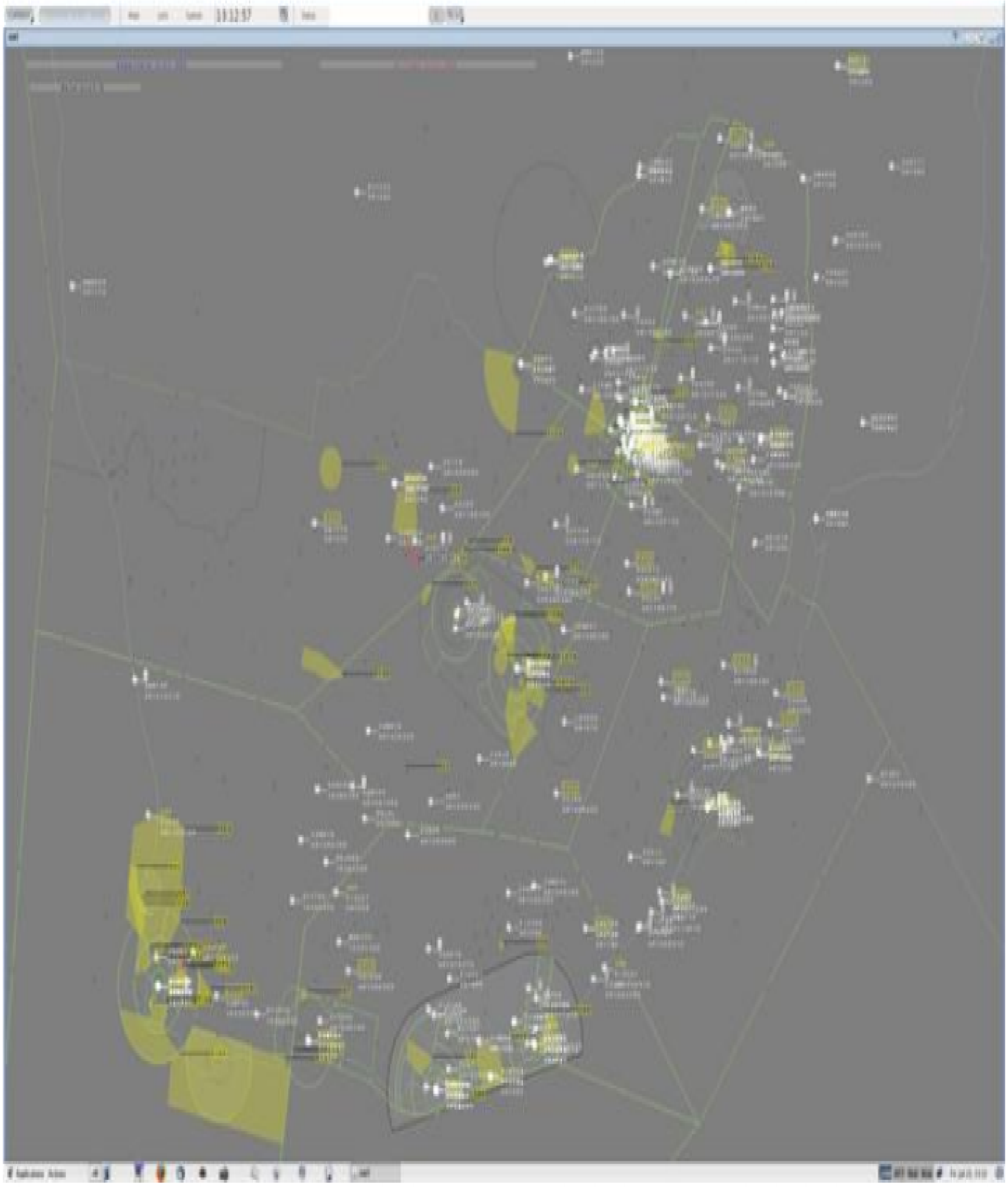


Figure 3.5: Airspace Management Tool used by CAMU operators (ATNS, 2015).

- provides situational mindfulness and report to the CAMU administrators utilizing the Airport Demand List (ADL) information, movement plan (including a blend of OAG timetable, IATA air terminal slots and flight data processor information);
- presents graphical and course of events presentation of aerodrome furthermore, airspace demand and capacity data;
- contains effective utilities for ground delay administration and examination

that considers brisk reaction to airspace imperatives by CAMU air traffic flow experts (ATNS, 2011, 2015).

For viable choice making, the CAMU administrators screen the aerodromes and Flow Constrained Areas (FCA) by viewing existing demand and limitations at those components and model the effects of potential traffic management initiatives (TMIs) with the aim of choosing the activity that yields the best answer for the present imperatives before it is finally implemented and reflected in the ADL (ATNS, 2011, 2015).

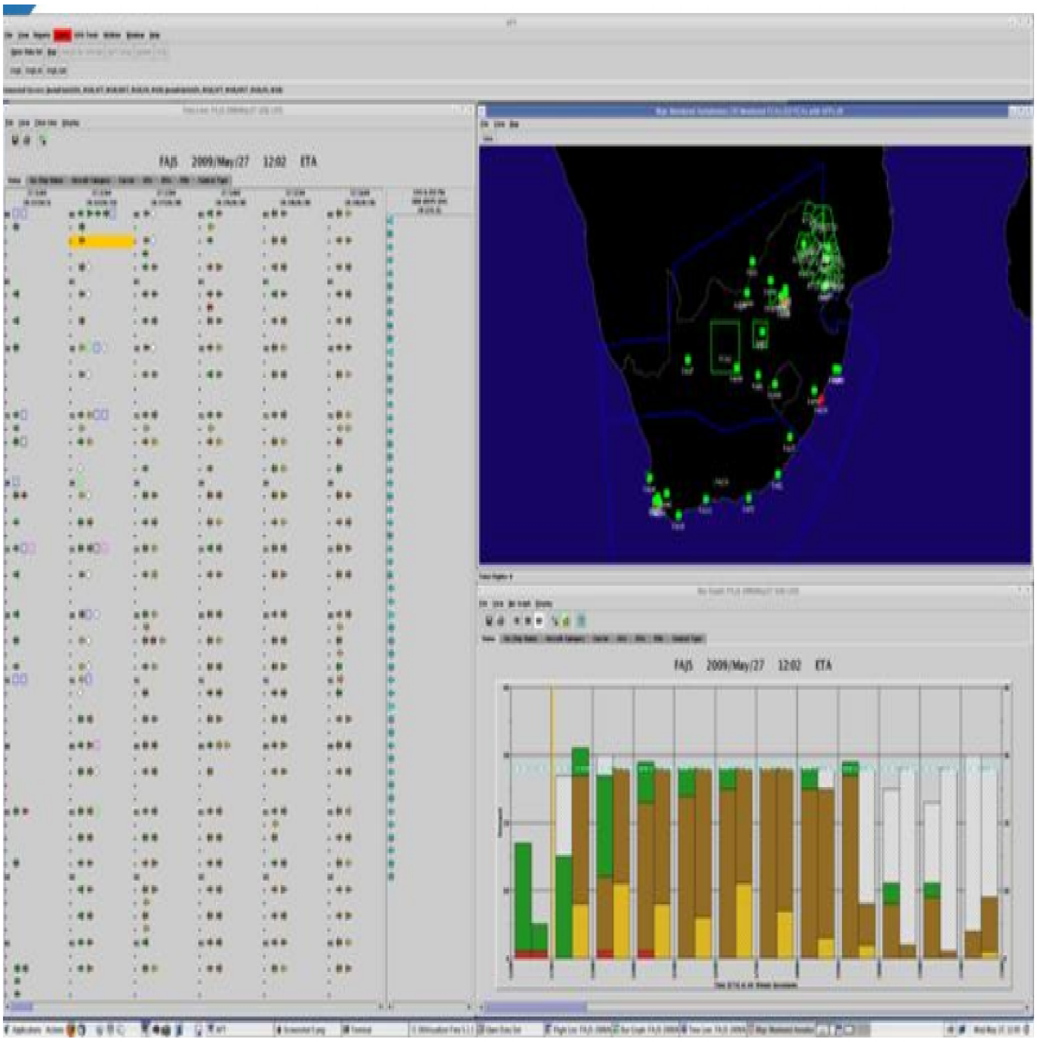


Figure 3.6: Airport Flow Tool (AFT) used by CAMU operators (ATNS, 2015).

6. **CAMU Web** (Figure 3.7) is an online investigation and strategic slot management tool (ATNS, 2011). The fundamental motivation for building up the web is to show current traffic initiative and related parameters; access the execution of cutting edge ATFM procedures progressively and give strategic slot management abilities through substitutions (ATNS, 2011, 2015). With the procurement of the online

interface, airlines have the choice of upgrading their utilization of accessible capacity. In addition, all airplane terminal partners have up-to-the-second shared situational mindfulness and booking (ATNS, 2015).

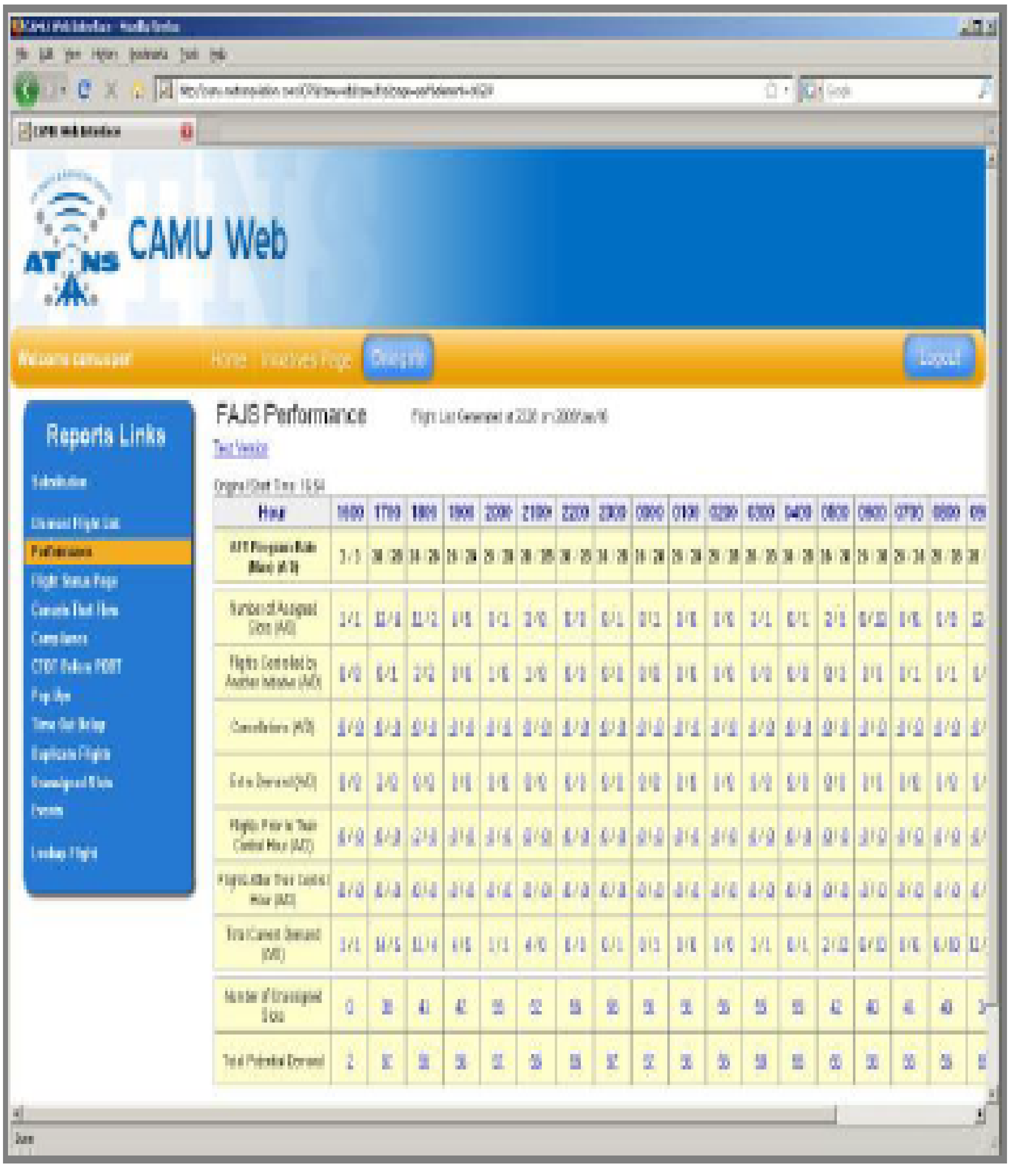


Figure 3.7: Screen shot of CAMU web interface (ATNS, 2015).

7. **METRON Traffic Flow, Thales Eurocat and FLOWCAT:** These software were developed by Thales with the support of Metron for the ATNS and CAMU for ATFM in South Africa. It was devised to meet the demand of airport and airspace users which in light of the 2010 World Cup, hosted in South Africa alongside a general increase in the country's tourism was set to increase. The flowchart in Figure 3.8 illustrates the ATFM system provided by Thales and Metrons for South

Africa ATNS while Figures 3.9 and 3.10 illustrates the Metron Traffic Flow .

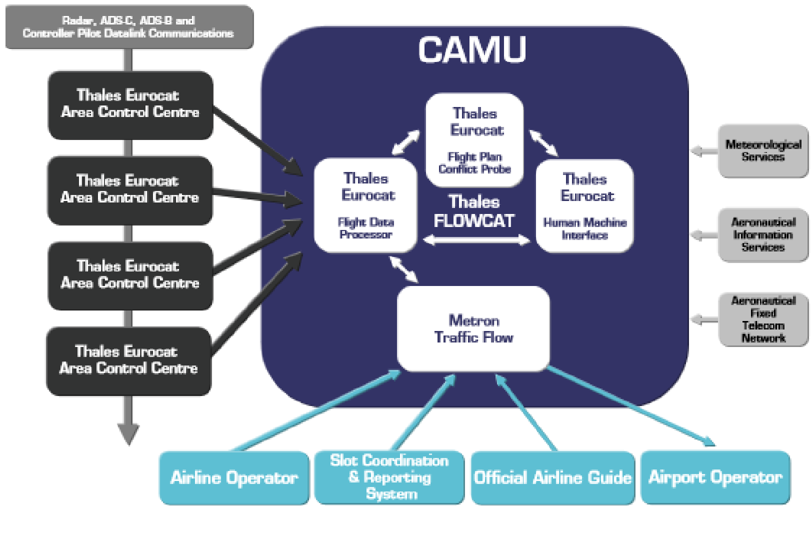


Figure 3.8: ATFM system provided by Thales and Metron for ATNS (ATNS, 2010).

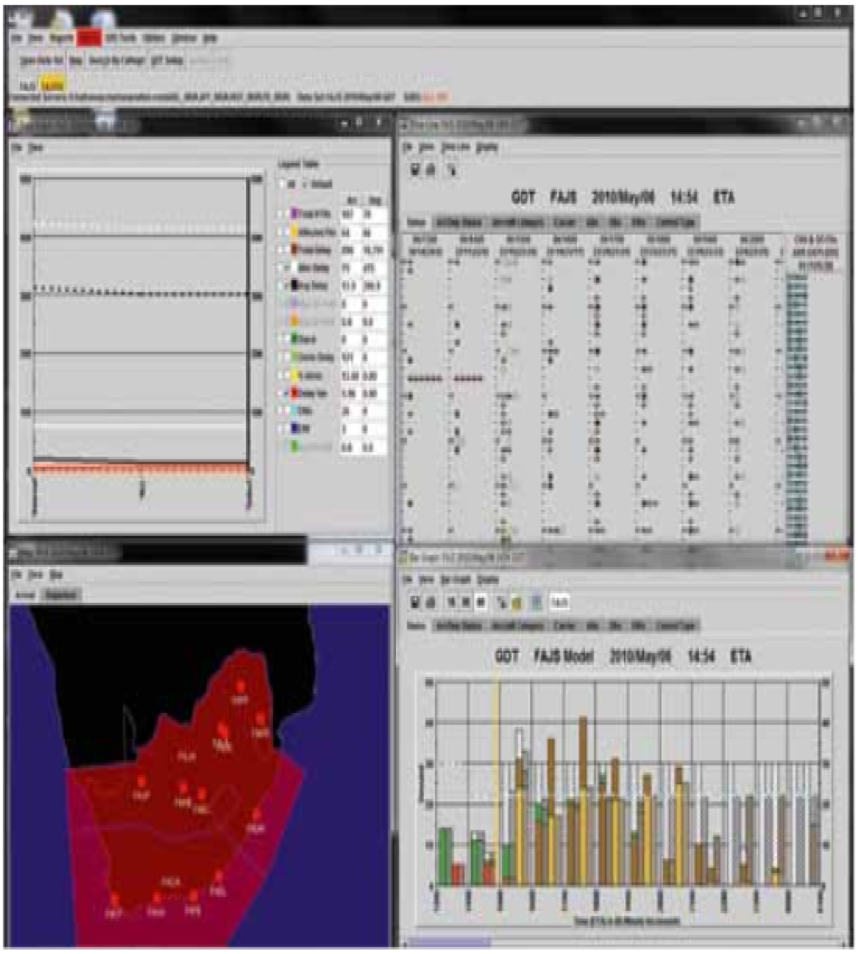


Figure 3.9: ATFM system provided by Thales and Metron for ATNS (ATNS, 2010).

Figure 3.10: ATFM system provided by Thales and Metron for ATNS (ATNS, 2010).

The system optimises the utilization of accessible airspace and airplane terminal resources by adjusting load and limit keeping in mind the end goal, that is to lessen delays, reduce clog and streamline the workload of air traffic controllers. The method includes the utilization of a Flight Schedule Monitor software to dole out landing slots to airplane in light of the available capacity and flight arrival times, including the delays in successive manner until demand equivalents the capacity (ATNS, 2010, 2011)

3.3 Optimisation Models for the ATFM Problem

As alluded to in the preceding section, there are several approaches or initiatives that are used by traffic managers to reduce the excessive demand before congestion occurs. These includes: *Ground Holding Approaches* – this is the strategic modification of landing and take off time of flights by assigning delays to flights selectively before departure which can be in the form of GS or GDP; *Airborne Control Approaches* – which includes spacing–miles-in-trails[2], speed controls[3], and vectoring[4]; *Rerouting approaches* – in which decisions are made on which flight routes an aircraft can traverse when it is known in ad-

[2] Spacing majorly controls the separation between aircraft moving in the same direction
[3] Speed control ensures "safety and efficient flow of aircraft by increasing or decreasing the speed"
[4] Vectoring can be seen as minor spatial deviation from a flight route (Vossen et al., 2012)

vance that the scheduled route is unusable due to reasons associated with poor weather conditions or other unforeseen factors. The distinguishing factor between these approaches is that *Ground Holding and Rerouting approaches are only applicable some hours in advance for effective control of the situation to be possible while Airborne approaches are used some minutes prior in order to tactically manage the situation at hand*. However, the miles-in-trails are an exception because they are employed to achieve a particular purpose as and when need arises. In the context of this review, these approaches are broadly classified into two major categories-Ground Holding and ATFM Approaches (Vossen et al., 2012).

ATFMP can be deterministic, stochastic, static and dynamic in nature depending on the aspect under consideration. Deterministic in the sense that *the capacities of the system are assumed to be known in advance* and stochastic in the sense that *capacities of the system are not known with certainty* (Agustín et al., 2009; Bertsimas and Patterson, 1998). On the other hand, ATFMP can be dynamic or static in the sense that the solutions are updated dynamically or not with time during the day. The optimisation model that are used to address the problem can be formulated either as dynamic, integer, stochastic programming and network-flow problem or any other method which are solved using different optimisation solvers and techniques like Lagrangian multipliers, relaxation methods, exact, heuristics and multi-objective algorithms and several others in order to get optimal solutions in good computational time. Thus, the review of related works on modeling and optimisation of ATFMP are classified based on these features:-

3.3.1 The Ground Holding Approaches

Most of the existing optimisation models that address ground holding approaches for solving ATFM are categorised under the airport capacity allocation models, which are further subdivided into two: the one that focuses on enhancing runway use through flight sequencing and another that concentrates primarily on capacity allocation amid discrete time interims among booked flights in an airport (Vossen et al., 2012). This is the most straightforward method for regulating the air traffic demand. The objective is to delay air plane on the ground as opposed to airborne, that is, exchanging airborne delay to the ground at the flight air terminal since it is more secure and less costly. This is quite advantageous because an aircraft will no longer have to fly extra distance in trying to avoid crowded regions of the airspace or flapping around a crowded airport. In essence, this also reduces fuel consumption. The rationale behind this methodology is to restrain the number of airborne aircraft at any given time with a specific end goal to decrease the controller workload and in addition fulfilling the capacity requirements of the aircraft arrival terminals. To achieve this, delays are selectively distributed to initially-scheduled aircraft departure times (Bertsimas and Patterson, 1998; Bertsimas and Odoni, 1997; Chaimatanan, 2014; Vossen et al., 2012).

There are two sub-classifications of problems when considering ground holding approaches: *Single Airport Ground Holding Problem (SAGHP) and Multiple Airport Ground Holding Problem (MAGHP).* The SAGHP is concerned with one destination airport at once in view of the presumption that reduction in capacity happens just at one airport in the framework while others have a boundless limit (Vossen et al., 2012). The Multiple Airport Ground Holding Problem (MAGHP) is a broadened variant of the SAGHP which incorporates the between relationship that exists between various airports (Agustín et al., 2009; Vossen et al., 2012). For SAGHP, the models do not consider constraints that are related to the en route sectors capacities rather it takes into consideration the arrival airport capacity as well as the departure airport capacity. The objective function of ground holding approaches is The target capacity of ground holding methodologies is to minimise the aggregate congestion and delay costs (with respect to the aggregate ground holding and to the airborne deferrals for the flights) while guaranteeing that the capacity limitations of the arrival airports are fulfilled (Agustín et al., 2009; Bertsimas and Odoni, 1997; Chaimatanan, 2014; Vossen et al., 2012). The presumptions ($A1 - A4$) of the model incorporates the accompanying:

A1) The *capacity of the arrival airport is known in advance with certainty.*

A2) *The capacity of both the departure airport and the airspace is unlimited.*

A3) *Alternative routes are not taken into account.*

A4) *The time horizon is discrete* (Agustín et al., 2009).

3.3.1.1 The Single Airport Ground-Holding Problem (SAGHP)

The SAGHP is one of the least difficult techniques for the ATFMP as it doesn't mull over the air sector capacity. As pointed out by Agustín et al. (2009), the SAGHP proposes answers for the issue of choosing the ideal planning for an airplane terminal, taking into the thought the confinements concerning the quantity of landing and takeoff operations that should be possible inside indicated time periods (Agustín et al., 2009)

Odoni (1987) was the first to formulate a model using mathematical programming related approaches for solving the SAGHP. Since then investigations have been going on, to decipher whether better strategies and methods can at all be developed to solve the problem (Balakrishnan and Chandran, 2015; Odoni, 1987). The strategy which was first studied in the United States of America (U.S.A) considers flight planning in real time with the aim of minimising congestion costs. They suggested an objective function based on the aircraft landing priority subject to probabilistic single-period airport capacities, and proposed a polynomial dynamic programming model to solve the fixed landing priority case and another optimal decision policy to solve a random priority, which depends on the chronological order of landing requests. Some of the assumptions ($B1 - B6$) of the basic SAGHP model found in literature are listed below

B1) *Airport is the only capacitated resource* in the problem.

B2) *Departure time and flight time are deterministic and known.*

B3) *Ground delay and air delay costs are known.*

B4) *Flight speed and alternative routes are not considered.*

B5) *Flight continuation is not permitted.*

B6) *Arrival advances in the schedule are not permitted* (Agustín et al., 2009; Vossen et al., 2012).

The given data of the problem include: *a set of flights, time periods, scheduled departure and arrival times to destination airports, a set of possible delay slots and associated delay costs, capacity of the arrival airports for the given scenario, the set of decision variables, which includes the arrival times to the destination airport, the ground delay attributed to each flight amongst other factors* (Agustín et al., 2009; Chaimatanan, 2014).

Related Works

Deterministic Models: For the deterministic case, the *airport capacities, departure time and flight time are deterministic*. Moreover, the problem can be formulated either as a minimum-cost network problem or using any other method. See Vossen et al. (2012); Terrab and Odoni (1993) and Richetta and Odoni (1993) for more details.

Presented below is the mathematical formulation of the basic model which starts with the definition of the input data sets. $F-$the *set of flights for an airport with capacity reduction*; $a_f-$the *scheduled arrival time* of a flight; $T-$the *time period* with t as the discretised time horizons; $M_t-$the *capacity of the airport* for each time period t; $X_{ft}-$the *decision variables* is defined as follows:

$$X_{ft} = \begin{cases} 1, & \text{if } \textit{flight } f \textit{ is assigned to time interval sector } t \\ 0, & \text{otherwise} \end{cases} \tag{3.3.1}$$

The expression $\sum_{t=a_f}^{T} (t - a_f)X_{ft}$ shows the *total ground delay* that was assigned to a particular flight f. It is important to note that *non linear ground delay cost functions* can be incorporated into the expression for the amount of ground delay that was assigned to each flight as $\sum_{t=a_f}^{T} g(t - a_f)X_{ft}$ where $g(.)$ is any arbitrary cost function. For example,

$\sum_{t=a_f}^{T}(t-a_f)^2 X_{ft}$ represents a quadratic ground delay cost function. Moreover, the ground delay costs can vary depending on the flights under consideration and thus can be weighed by some constant, say, C_f (Vossen et al., 2012).

The generalised formulation− the objective function and set of constraints is given by

$$Min \sum_{f \in F} \sum_{t=a_f}^{T} C_f g(t-a_f) X_{ft}$$

subject to:

$$\sum_{t=a_f}^{T} X_{ft} = 1 \quad \forall\ f \in F. \tag{3.3.2}$$

$$\sum_{f \in F\ :a_f \leq t} X_{ft} \leq M_t \quad \forall\ t \in T \tag{3.3.3}$$

$$X_{ft} \in \{0,1\} \quad \forall\ f \in F, t \in T. \tag{3.3.4}$$

The formulated problem can be solved as a linear IP problem via LP relaxation since *the objective function is linear and the constraint matrix is totally unimodular* (Vossen et al., 2012). Hence, integer solutions are always guaranteed. The constraints ensures that one time interval is assigned to a flight and the capacity of the airport is not exceeded. The above formulation also shows that the proposed model is of polynomial computational complexity (Vossen et al., 2012)

Terrab and Odoni (1993) additionally introduced the deterministic model for SAGHP with the aim of minimizing the aggregate cost of ground holding for the set of flights and utilizing a linear cost function with specific parameters to represent the delay cost of each flight. In any case, Hoffman et al. (1999) is a variant of the model as it considers '*banking constraints*' with the condition that one or more groups of flights must arrive within pre-determined time windows and this depicts the current practice as most airlines '*schedule banks of operations at their hub airports* (Terrab and Odoni, 1993; Hoffman et al., 1999). The rationale for including the banking constraints by Hoffman et al. (1999) was to keep flights of each banks incidentally gathered. Be that as it may, this really influenced the computational execution of the model as far as the time taken to get optimal solution since the LP relaxation does not give integer solution always.

Aside from allocating ground delays to approaching flights (which has dependably been the situation for most SAGHP models), others like that of Gilbo (1993) presented models that relegates ground delays to both landing and takeoff movement and augmented it in Gilbo (1997) to incorporate arrival and takeoff fix capacities at an airplane terminal close by its runways Gilbo (1993, 1997). A comparable issue was that of Dell'Olmo and Lulli (2003), which however was unravelled efficiently utilising a dynamic programming algorithm to acquire optimal solutions. In all the deterministic models introduced, the arrival and departure capacities of an air plane terminal are associated. The reliance between these limits happens when there are intersection runways or when two runways are excessively near each other or when a runway is utilised for both landing and departure operations. This is usually captured with the concept of capacity envelope and runway allocation[5] (Vossen et al., 2012; Bertsimas and Patterson, 1998).

Stochastic Models: Aside reductions in airport capacities, bad weather conditions also affects the departure capacities of an airport. This knowledge led to the modeling of the SAGHP using stochastic approaches as it captures cases of uncertainty.

Stochastic programming approaches were first used to solve the SAGHP by Richetta and Odoni (1993). The problem was formulated as a one-stage stochastic linear program with static weather scenarios, which were not updated with time. However, the problem sizes were reduced by including only a small number of weather scenarios and aggregating flights into cost classes. The problem can be simplified by assuming a constant air delay cost in order to achieve a reasonable solution time.

Dynamic Programming Models: Terrab and Odoni (1993) extended the deterministic version to stochastic version. The authors developed an exact dynamic programming model based on the work presented in Andreatta and Romanin-Jacur (1987).

Richetta and Odoni (1994) improved their previous study by providing a partially dynamic multistage stochastic integer programming formulation with recourse actions while maintaining previous assumptions. That is, making delay decisions at each stage of the updated weather forecast and no decision once the delay has been assigned (Chang, 2010).

The dynamic ground holding model, Mukherjee and Hansen (2007), is an improvement of Richetta and Odoni (1994), in the sense that it allows for revision of the ground delay decisions based on the latest information for flights that are yet to depart once the delays has been assigned.

Liu et al. (2008) applied scenario-based approaches to the models in Mukherjee and Hansen (2007) as well as Ball et al. (2003) by studying the capacity scenarios based on the historical data provided. The study reveals that certain scenarios which follow a tree-like structure possess same capacities.

Static Programming Models: Considering the uncertainty in the capacity of arrival airport, Richetta and Odoni (1994) were the first to suggest a *Static Stochastic Integer Programming Model* in order to solve the problem albeit its modification by Ball et al. (2003) to find optimal number of scheduled aircraft arrivals during different time intervals. To assign delays to departing flights, decisions are made at the commencement of the scheduling process and cannot be reversed once assigned. This short-fall was later

[5] Capacity envelope is a "piecewise linear convex functional relationship that shows the interdependence between arrival and departure capacities of airports" (Vossen et al., 2012).

addressed in Richetta and Odoni (1994) via a new formulation of a *Multistage Stochastic IP with Recourse*. See Agustín et al. (2009), Vossen et al. (2012) and Barnhart et al. (2012) for more details.

3.3.1.2 The Multiple Airport Ground-Holding Problem (MAGHP)

This methodology is an extended version of the previous methodology which includes the inter-relationship that exists between network of airports. The contrast between MAGHP and SAGHP is that MAGHP contemplates the proliferation of delays in the network of airports as aircraft perform sequential flights, that is optimization of the ground delays allocated to flights in a manner that a delay on a given flight section can spread crosswise over to down fragments own by the same aircraft (Agustín et al., 2009; Bertsimas and Odoni, 1997; Chang, 2010). Although, the MAGHP approaches considers arrival and departure capacities in the network of airports, *en-route airspace capacity constraints* are not considered. Moreover, most models for the MAGHP do not carter for flights whose departure or arrival airport does not belong to the set of airports under consideration. One of the real target of MAGHP is to discover an arranging model that minimises the aggregate cost of delays, airborne comprehensive subject to the airport capacity and flight network imperatives (Agustín et al., 2009). This can be achieved by simulating various alternatives of the said infrastructures. The vast majority of the MAGHP models are deterministic in nature. The essential presumptions ($C1 - C7$) of the models are recorded beneath

C1) *The airport capacities are deterministic function of time and are known before hand with certainty.*

C2) *Air sector airspace capacity is unlimited.*

C3) *There is no limit on the ground holding upper bounds as well as that of the air borne delay.*

C4) *Flight cancellation are considered on a partial basis based on the previous assumption.*

C5) *Flight speed and alternative routes are not considered.*

C6) *Flight continuation is permitted.*

C7) *The turnaround time* [6] *is always known for continued flights* (Agustín et al., 2009; Bertsimas and Patterson, 1998).

The datasets for the problem include: *a set of arrival and departure airports, set of airports by independence, scheduled departure times for flight, scheduled arrival times to destination*

[6] A *turnaround time* is the “slack time between the arrival of a flight to an airport and the departure of given flights from the same airport”.

airports, air and ground holding delay costs, the departure and arrival airports capacities, decision variables, which includes assignment and delay decision variables (Agustín et al., 2009; Bertsimas and Patterson, 1998; Chaimatanan, 2014).

Related Works

Deterministic Models: Vranas et al. (1994) were the first to propose a deterministic optimisation for multi-airport ground holding problem which was formulated as an integer program (IP). The computational performance of their model revealed that the run times to get optimal solution when solving real-life instances are very high. The authors also presented a stochastic optimisation model of MAGHP considering uncertainty in the airport capacities. However, an alternative formulation was presented by Bertsimas & Patterson Bertsimas and Patterson (1998) which performed better computationally when compared with that of Vranas et al. (1994). The performance was as a result of the facet-defining constraints that were used in the formulation. This formulation was extracted from the model they presented for the airspace capacity allocation problem.

Andreatta and Brunetta (1998) made a comparison of three distinct models for the MAGHP. The models they considered were that of Andreatta and Tidona (1994)−where insufficient arrival capacity causes a congestion problem based on the assumption of infinite departure capacity; Vranas et al. (1994) and MAGHP case of Bertsimas and Patterson (1998)−where the 0-1 variables takes *the value 1 if and only if a flight f has arrived by time t that is it arrives at time t or earlier respectively* (Agustín et al., 2009; Andreatta and Brunetta, 1998). The models were validated using a set of seven test bed instances in a bid to ascertain their computational performance which was shown to be good.

The deterministic models developed for the MAGHP can be extended to account for several variations. See Bertsimas and Patterson (1998) and Navazio and Romanin-Jacur (1998) for more details.

Agustín et al. (2009) pointed out that the proposed methods by Andreatta and Tidona (1994), Vranas et al. (1994) and several others have been applied to solve different problems simulated at the Drapper Laboratory, MITRE Corporation using FAA data and Boston Logan Airport respectively (Agustín et al., 2009).

Recently, Zhang et al. (2007) solved the MAGHP using a stochastic search approach by *developing a co-evolutionary Genetic Algorithm.* The model was validated with real data obtained from the *Beijing, Shangai and Guangzhou traffic control centres* of the China Aviation Administration which was considered good computationally because of the characteristic features of the Genetic Algorithm. It therefore created more research opportunities for modelling MAGHP via stochastic search techniques (Agustín et al., 2009; Zhang et al., 2007).

3.3.2 The Basic Air Traffic Flow Management Problem

ATFMP is one of the methods that is used to address the air traffic delay and congestion problem. The approach involves control and management of the air traffic flow within the airspace sector capacity apart from ground holding and air borne delay. Besides regulating the departure time of flights, ATFM also *seeks optimal arrival times to each airspace sector, taking into account airspace capacity constraints* (Chaimatanan, 2014). This methodology depicts clearly what ATFMP is all about in the sense that it tries to solve realistic instances that are more complicated than those solved using the past methodology. This is because the former only applies where airport congestion problems occur frequently while the latter applies not only when congestion problems are confined to airports but also when en route sector congestion occurs. Most of the optimisation models that takes into consideration airspace capacity constraints alongside ground holding and air borne delays are categorised under the airport capacity allocation models (Agustín et al., 2009; Bertsimas and Odoni, 1997; Chaimatanan, 2014).

Other sub-categories of problems considered in literature under ATFM includes *ATFM Rerouting Problem (ATFMRP), ATFM Rerouting Problem with Uncertainty and many more.*

The general assumptions ($D1-D5$) of the *basic ATFM model* problem includes but is not restricted to the following:

D1) *The arrival and departure airports capacities as well as the airspace capacities are known in advance with certainty* (Bertsimas and Patterson, 1998).

D2) *Flight speed is not taken into consideration.*

D3) *Alternative routes are not considered as a control option.*

D4) *Flights are continued in such a way that the turnaround time is known for the continued flights.*

D5) *Flight cancellation is permitted although no much decisions are made since it is implicitly considered continuation is allowed* (Agustín et al., 2009; Bertsimas and Patterson, 1998; Chaimatanan, 2014).

The objective function is *to minimise the total cost of delays assigned to flights while satisfying the airport and the airspace capacity constraints* (Bertsimas and Patterson, 1998; Chaimatanan, 2014). The problem input sets and parameters include, a *set of flights, airports, continuing flights, sectors, time periods, scheduled departure and arrival times, associated ground delay and airborne delay costs per unit of time, decision variables which includes includes the ground delay times, the air delay times, the time at which each* *flight arrives at a given airspace or at a given waypoint* (Bertsimas and Patterson, 1998; Chaimatanan, 2014).

According to Bertsimas et al. (2011), *the ATFMP represents the air traffic network by a directed graph with nodes*[7] *and arcs*[8]. For instance, an arc from node i to node j exists if i and j are adjacent sectors such that an aircraft can reach sector j immediately after flying through sector i. Given a flight, f, the set of sectors that follow sector i is denoted by L_i^f while the set of sectors that precede sector j is denoted by P_j^f (Bertsimas et al., 2008; Chaimatanan, 2014). See Fig. 3.11 for illustration:

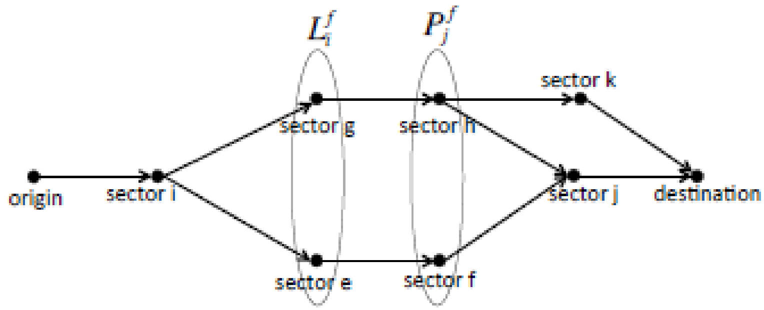


Figure 3.11: Illustration of possible routes from O-D airport (Bertsimas et al., 2008, 2011).

3.3.1 Remark. *An arc is required to be fully contained within a sector (i.e. it cannot cross sector boundaries). In other words, a sector could be viewed as a collection of arcs.*

3.3.2 Remark. *Nodes are introduced at the sector boundaries to ensure that arcs do not cross sectors, although a node could be present in the interior of a sector.*

3.3.3 Remark. *Bertsimas et al. (2011) noted that sectors followed by more than one sector are called forks while those sectors preceded by more than one sectors are called joints. In Fig 3.11, sectors i and h are forks while sector j is a joint.*

Related Works - Air Traffic Flow Management Problem

Deterministic Models: Helme (1992) formulated the National Air Space (NAS) traffic management problem considering both the airport and air sector capacity constraints. The problem was formulated as *a multi-commodity minimum-cost flow on a space-time network* which deals with aggregate flows of flight instead of individual flights based on the assumption that flights route are pre-determined before departure.

The computational performance of the model was weak when compared to others but performed better when compared to models formulated as a single-commodity flow network.

[7]A *node* can be either a physical location, corresponding to a region in the airspace, or a set of capacitated components, that is, airports and airspace sectors Bertsimas et al. (2011)

[8]An *arc* is a directed segment connecting two nodes which also represents the sequence relations while a *sector* is a contiguous region of an airspace. See Section 3.1.

Lindsay et al. (1993) formulated a deterministic IP model in order to assign ground and airborne delays to individual flights taking into considerations the capacity constraints of both airport and airspace (Agustín et al., 2009; Balakrishnan and Chandran, 2015; Lindsay et al., 1993).

A deterministic optimisation model that depicts the problem of ATFM in European (EU) environment was formulated by Lulli and Odoni (2007) where they illustrated the intricacy of the ATFM in European locales, the benefits of doling out airborne holding delays to a few flights deliberately and the issues of equity that can emerge as a consequence of the interactions among traffic flows. One critical angle that the proposed EU ATFM model tended to was the contentions that might emerge between the objectives of efficiency of and equity−fairness (Bertsimas et al., 2011; Lulli and Odoni, 2007).

It is important to note that neither rerouting of flights nor speed control were considered in all the afore-mentioned models as a control action for the ATFMP.

Akgunduz et al. (2013) introduced a mixed 0-1 IP model in order *to minimise the travelling time, operating cost, air pollution which are subjected to some constraints*. The decision variables for the model include *speeds, departure and arrival times for each flight, the arcs and nodes but in a 3D-mesh network*. The problem is then solved by an exact deterministic method but on instances involving not more than 10 flights and the was computationally efficient (Akgunduz et al., 2013; Chaimatanan, 2014).

Bertsimas and Patterson (1998) proposed a 0-1 IP deterministic model for the ATFMP. They defined the route between the origin-destination of each aircraft as a *predefined set of en-route sectors*. The proposed model determines *the optimal departure time, as well as, sector occupancy time* (the time expected for each aircraft to stay in a sector) for each aircraft (Bertsimas and Patterson, 1998; Bertsimas et al., 2011). The definition of the decision variables with "*by*" rather than "*at*" and the transformation done from the standard decision variable to the variable used were critical for the models excellent performance in addition to the facet-defining constraints and the polyhedral structure of the basic LP relaxation problem. They illustrated the complexity of the problem by proving that it is NP-hard. The LP-relaxation of the formulation gives integer optimal solutions in most cases. The general model formulation is illustrated below (Bertsimas and Patterson, 1998).

For the input data sets, $K, F, J, T, \mathcal{C} := \{(f, f') : f' \text{ is continued by } f\}$ represents the set of airports, flights, sectors, time periods which are discretised into time horizons and set of continued flights respectively. For any given flight f, N_f represents the cardinality of sectors on a flight's route; $P(f, i)$, the i^{th} sector in the flight's path while $P_f = (P(f, i) : 1 \leq i \leq N_f)$ represents the predefined route for the flight with $P(f, 1)$ and $P(f, N_f)$ been the origin and destination airports. The capacities for the departure and arrival airports for each time interval is denoted by $D_k(t)$ and $A_k(t)$ respectively taken over all airports and time periods. The number of aircraft permitted to be in a particular sector at a particular time period is denoted by $S_j(t)$. This restriction is referred to as the "**en route sector capacities**" (Bertsimas and Patterson, 1998). l_{fj} represents the minimum number of time spent by the flight in each sector. a_f, d_f, s_f, r_f, g_f represents the scheduled arrival time, scheduled departure time, turnaround time, airborne delay and ground delay assigned to a flight respectively. c^a_f and c^g_f denotes the cost of delaying a flight in the air and on the ground per unit of time. Moreover, for each flight, T^j_f, the set of feasible time periods (that is set of feasible time window for each flight that is an indicator of the time that a flight can occupy a resource along its flight route) for flight f to arrive to sector j is defined as $T^j_f = [\underline{T}^j_f, \overline{T}^j_f]$ where $\underline{T}^j_f$ and $\overline{T}^j_f$ represents the first and last time period in the set T^j_f respectively without detailed description. It is important to note that the set of feasible time periods are calculated sets in preprocessing defined for the purpose of reducing the size of the problem formulation.

The decision variables are defined as

$$w^j_{f,t} = \begin{cases} 1, & \text{if flight } f \text{ arrives at sector } j \text{ by time } t \\ 0, & \text{otherwise} \end{cases} \tag{3.3.5}$$

The objective function is given by

$$Min \sum_{f \in F} \left(c^g_f g_f + c^a_f r_f \right), \tag{3.3.6}$$

where

$$g_f = \sum_{t \in T^k_f : k = P(f,1)} t(w^k_{f,t} - w^k_{f,t-1}) - d_f \tag{3.3.7}$$

and

$$r_f = \sum_{t \in T^k_f : k = P(f,N_f)} t(w^k_{f,t} - w^k_{f,t-1}) - a_f - g_f \tag{3.3.8}$$

Subject to

$$\sum_{f : k = P(f,1)} (w^k_{f,t} - w^k_{f,t-1}) \leq D_k(t) \quad \forall\, k \in \mathcal{K}, t \in \mathcal{T} \tag{3.3.9}$$

$$\sum_{f : k = P(f,N_f)} (w^k_{f,t} - w^k_{f,t-1}) \leq A_k(t) \quad \forall\, k \in \mathcal{K}, t \in \mathcal{T} \tag{3.3.10}$$

$$\sum_{f : j = P(f,i), j' = P(f,i+1), i \leq N_f} (w^j_{f,t} - w^{j'}_{f,t}) \leq S_j(t)\ \forall\, j \in \mathcal{J}, t \in \mathcal{T} \tag{3.3.11}$$

$$w^{j'}_{f,t+l(f,j)} - w^j_{f,t} \leq 0 \quad \begin{cases} \forall\, f \in \mathcal{F}, t \in \mathcal{T}^j_f : \\ j = P(f,i), j' = P(f,i+1), i \leq N_f \end{cases} \tag{3.3.12}$$

$$w^k_{f,t} - w^{k'}_{f',t-s_f} \leq 0 \begin{cases} \forall\ (f', f) \in \mathcal{C}, t \in \mathcal{T}^k_f : \\ k = P(f, i), k' = P(f', N_f) \end{cases} \tag{3.3.13}$$

$$w^j_{f,t} - w^j_{f,t-1} \geq 0 \ \ \forall\ f \in \mathcal{F}, j \in P_f, t \in \mathcal{T}^j_f \tag{3.3.14}$$

$$w^j_{f,t} \in \{0, 1\} \ \ \forall\ f \in \mathcal{F}, j \in P_f, t \in \mathcal{T}^j_f \tag{3.3.15}$$

Since the the departing and arriving airports are defined as the starting and ending sector of each flight respectively, equations (3.3.7) and (3.3.8) implies that the time units of ground delay assigned to a flight can be expressed as *the actual departure time minus the schedule departure time* while the amount of air borne delay assigned to each flight is expressed as *the actual arrival time minus the scheduled arrival time minus the amount of ground delay assigned to the flight* (Bertsimas and Patterson, 1998).

The first three constraints are the *capacity constraints* satisfying the departure, arrival and sector capacities while the subsequent three constraints are the *operational or connectivity constraints* satisfying sector, flight and time connectivity respectively. They are facet-defining and that is why the model performs very well computationally. The last two constraints ensure that the decision variables are binary variables and non-negative. Most of the models presented beforehand are variations of the Bertsimas and Patterson (1998) model formulation. The removal of the sector capacity constraints from the BATFMP formulation will correspond to a MAGHP formulation. Moreover, if flight continuity constraints is also removed and the set of airports, K, is assumed to be a singleton set, then the resulting formulation will be a SAGHP. The authors presented a MAGHP formulation which they compared the performance of the model to that of Vranas et al. (1994) and it performed excellently. One presumption of the model is that flight network requirements should have been fulfilled from the earlier in view of planned aircraft connections. This, however is not generally the situation practically speaking because of the vicinity of hub and spoke systems which energize the utilization of bank of flights (Bertsimas and Patterson, 1998; Bertsimas and Gupta, 2011).

3.3.2.1 The Air Traffic Flow Management Rerouting Problem

Related Works−ATFM Rerouting Problem

Bertsimas and Patterson (1998) illustrated how the models can be extended to incorporate *the dependence of airport runway capacity of departures and arrival, handle banks in the spoke system, hub connectivity and multiple connections* using the concept of capacity envelopes and runway allocation and modifying the flight connectivity constraint respectively. They also presented two possible approaches namely, *the path and sector approaches, which illustrates how their model can be modified to include rerouting options when there is drastic drop in the capacity levels* (Bertsimas et al., 2008; Bertsimas and

Patterson, 1998).

The path approach decides which *route to fly among a set of possible routes for each aircraft* while the sector approach determines which *sector to enter next for each flight on its route* (Bertsimas and Patterson, 1998). This is where the idea of graph theory and partially ordered set comes in as the traffic network is always assumed to be an acyclic digraph.

Deterministic Models: Bertsimas and Patterson (2000), considered rerouting at a broader level by presenting a model that addresses routing as well as scheduling decisions. The problem was formulated as a *dynamic, multi-commodity, integer network flow with side constraints* (Bertsimas and Patterson, 2000). To solve the problem, they generated aggregate flows of flights and using a Lagrangian relaxation, they solved the LP problem with the constraints relaxed into the objective function. The aggregate flows were later decomposed into sets of individual flights by applying rounding heuristics which are random in nature. The optimal and near optimal solutions of the individual flight routes were obtained by solving an integer packing problem. The computational performance was inadequate for addressing large-scale problems in real life situations (Agustín et al., 2009; Balakrishnan and Chandran, 2015; Vossen et al., 2012).

Bertsimas et al. (2008) addressed this limitation by presenting another model that combines the flexibility and properties of their previous models to *optimise for each flight; the departure time, the flight plan, the time needed to cover each sector, and the arrival time, taking into consideration the capacity of the air traffic management system* (Bertsimas et al., 2008). Later, it was discovered that the sector capacity constraint of the model has a computational problem and this has been addressed in a more recent work by the same authors in Bertsimas et al. (2011) by introducing a maximum function in the sector capacity constraint. However, as a result of the maximum function that was introduced, the model contains non-linear constraints giving rise to a non-linear model. The proposed model is an extension of Bertsimas and Patterson (1998) formulation to account for rerouting on an acyclic network. The problem was formulated as an IP model taken into consideration all the stages of each flight as optimisation approaches were used *to allocate ground delays and rerouting options taking into account airspace sector capacity constraints*. They introduced new inequalities for nodes in the network with in-degree or out-degree equal to 1, which are valid and were shown to be facet-defining for the problem. Another advantage of their model is the fact that the initial variables were maintained and the approach for defining the routes based on the principles of graph theory and partially ordered sets (POSET) (Bertsimas et al., 2008).

The origin-destination route is represented by *digraphs* and is assumed to be *acyclic*[9]

[9] A *graph*, G, is a representation of a set of points called vertices, V, in which the pairs are connected (joined) by lines or links called edges, E. More precisely, a graph is denoted as $G = (V, A)$ or $G = (V, E)$. A graph is said to be a *diagraph* if the edges, elements of E, have a direction associated with them. A digraph is *acyclic* if the graph have no directed cycles (Chinneck, 2006)

without loss of generality. The authors set local conditions that ensures that the flights follow exactly one route which is similar to *the mass balance constraints in network flow model* (Bertsimas et al., 2008). The additional notations which they introduced that formed part of the new constraints are the same as the ones illustrated in Fig 3.11. The notations were used to describe the routing conditions. The computational performance of model was very good when applied to regional-sized and nation-sized instances due to the polyhedral structure, the facet-defining constraints and the new valid inequalities that were included. Moreover, all the works described above and presented in Bertsimas et al. (2008, 2011); Bertsimas and Patterson (1998) rely on *branch-and-bound algorithms,*[10] *which considers all feasible points of the optimisation problem implicitly in order to provide optimal solutions to the ATFMP*(Bertsimas et al., 2008; Chaimatanan, 2014).

In fact, these models have been identified as the best optimisation models for addressing the ATFMP as all other models are variants of the basic model. See Bertsimas and Patterson (1998, 2000); Bertsimas et al. (2008, 2011) for more insights on the computational results of the models.

Balakrishnan and Chandran (2015) as well as Agustín et al. (2012a) addressed both the deterministic and stochastic cases of ATFMRP. For the deterministic case, Balakrishnan and Chandran (2015) presented an integer programming approach for solving large scale ATFMP. Their model considers almost all the control actions and their representation is on a standard node-link network which is unrestricted. To determine the best trajectories in space and time satisfying the network and flight connectivity constraints, the authors used column generation approach in which they solved the LP relaxation problem, applied rounding heuristics to estimate the optimality. The master problem is finally solved using CPLEX and a scalable parallel implementation of their approach is used to solve nation-size instances which reveals that their model is the computationally efficient and their approach is good for real time implementation.

Agustín et al. (2012a) formulated a mixed 0-1 IP model for the ATFM which also considers almost all the control actions including *bounds and penalisation for the deviations in the current travel time from the scheduled arrival time of the flights to the nodes in the sectors.* Rerouting in their work was defined *exactly in the same way as a scheduled route* that is defining links that joins two nodes but when a node has different possible links, it is then forced to take one and go on the chosen route, thereby penalising the alternatives (Agustín et al., 2012a). The proposed route representation allows the ATFM model to be represented by a collection of subgraphs $G_f = (N_f, A_f)$, where the node set, N_f represents the airports and waypoints overflown by flight f, and the arc set, A_f, connects the nodes for flight f (Agustín et al., 2009). The decision variables, $x^t_{f,m,n}$, takes value 1 *if flight* f *is scheduled to arrive at node* n *at time* t *through arc* (m, n) ***by*** *the end of time period* t*, and takes value 0 otherwise* (Agustín et al., 2012a). The proposed model likewise

[10] See Section 2.1.4.1

considers equity between flights notwithstanding minimising the aggregate ground delay and airborne delay costs. The major contrast between the proposed model and different models existing in literature is that the configuration depends on the arcs (links) of the routes and not exclusively on the nodes and does not require the execution of the branch-and-bound stage in order to obtain an optimal solution in any given instances (Agustín et al., 2012a). Likewise, the model acquainted new objective functions to be optimised in addition to the standard objective function found in literature (Chaimatanan, 2014; Agustín et al., 2012a).

3.3.2.2 Air Traffic Management Rerouting Problem Under Uncertainty

As mentioned earlier, ATFMRP is more concerned with real life situations where flights can be rerouted whenever congestion problems arises notwithstanding unforeseen changes that can also occur in the airport and air sector capacities (and these vary significantly with weather conditions). These situations are difficult to predict with certainty in advance thus the need for approaches that can work under uncertainty in order to account for these unforeseen changes. ATFMRP with uncertainty is a variation of the previous methodologies and depicts the current practice because of the generic uncertainty incorporated in the model parameters which can then be solved using stochastic methodologies. These can also be *static or dynamic*. They are considered to be static if the solution holds firm from the start of the planning horizon and dynamic if the solution is updated throughout the planning horizon (Agustín et al., 2009; Balakrishnan and Chandran, 2015).

Stochastic optimisation is one of the options to consider when modeling weather uncertainties because it *allows representation of uncertainty by means of set of possible scenarios that can arise* (Agustín et al., 2012b).

3.3.4 Definition. *A **scenario** is a realisation of the uncertain parameters along the periods of the given time horizon while a **scenario group** for a given period is the set of scenarios with the same realisation of the uncertain parameters up to the period (Agustín et al., 2009, 2012b).*

Agustín et al. (2009) pointed out that each of these scenarios weights according to the occurrence of its certainty known to the decision-maker as well as the proposed solution must be such that all scenarios are considered and it does not depend on any one of them in particular (Agustín et al., 2009, 2012b).

Agustín et al. (2012b) outlined the different approaches involved in stochastic optimisation; *decomposition, Lagrangian decomposition, disjunctive decomposition, stochastic branch-and-cut and Branch-and-Fix Coordination* and noted that the objective functions considered by these approaches can be categorised as (*neutral risk*)[11], (*risk aversion*)[12]

[11] neural risk is "maximising the expected objective function value over the scenarios"

[12] risk aversion is "maximising a composite mean-risk function-included by the expected objective func-

and (*robust optimisation*)[13] (Agustín et al., 2009).

The literature on solving stochastic cases of ATFMP is typically limited to optimising flows into a single capacitated airport under probabilistic scenario-tree forecasts of airport capacity following the difficulty encountered when trying to solve large-scale deterministic ATFMP (Ball et al., 2003; Mukherjee and Hansen, 2007; Richetta and Odoni, 1994). Although most of the stochastic optimisation approaches dealing with weather uncertainties are scenario based, some researchers have also modelled *weather uncertainty and evolution as a stationary Markov chain* while others used different methodology from mathematical, dynamic or stochastic programming techniques. Considering the illustration of the scenario tree in Fig 3.12 as given by Agustín et al. (2012b), the following descriptions are an interpretation of the notations used. $\mathcal{T}-$the *set of consecutive stages along time horizon*; $\Omega-$ the *set of scenarios*; $\mathcal{R}-$ the *set of scenario groups in such a way that there is a tree where* $\mathbb{R}$ *is the set of nodes*; $\mathbb{R}_t-$ *set of scenario groups in stage* t for $t \in T$; Ω_r- *set of scenario groups in* r for $r \in \mathcal{R}$. $t(r)$, $a(r)$, $\mathcal{V}_r$, $\mathcal{V}^r$ and ω_r represents the *time period where* r *belongs; immediate ancestor node of node* r*; set of scenario groups to group* r *including* r*; set of successors of scenario group* r *and weighted factor representing the associated likelihood with* r *respectively* (Agustín et al., 2009, 2012b).

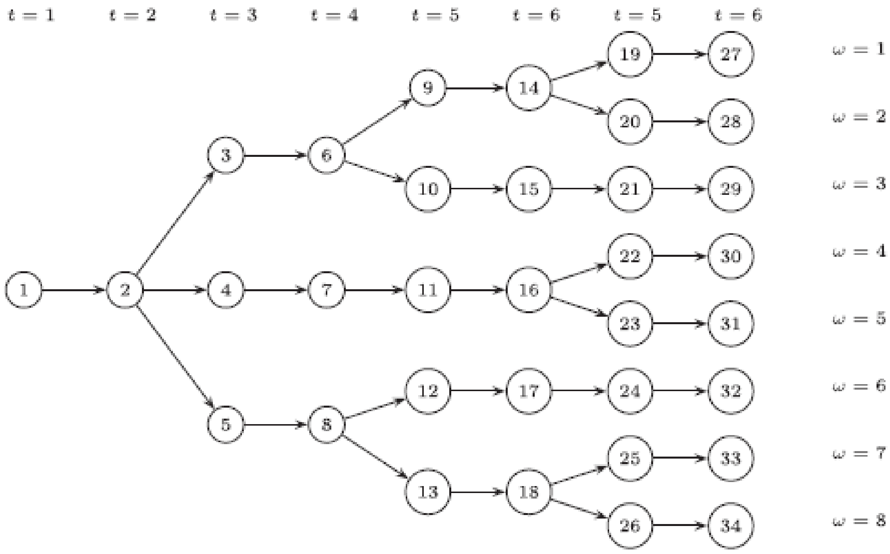
Each node in the figure denotes *a point in time where a decision can be made*. Moreover, contingencies occurs once a decision is made with the relevant information associated to these contingencies are known at the beginning of the period. For example, in the illustrated figure, period $t = 3$, contingencies occurred three (3) times (Agustín et al., 2009, 2012b). The information structure is visualised as a *tree*, where each root-to-leaf path represents a specific scenario corresponding to a *realisation of sets of uncertain parameters* with the scenario groups associated with each node in such a way that two scenarios belongs to the same group in a given period provided that they have the same realisations of uncertain parameters up to that period (Agustín et al., 2012b).

3.3.5 Definition. *A period is a set of consecutive time units where the realisation of the uncertain parameters take place while a stage is a set of consecutive periods where the realisation of the uncertain parameters take place (Agustín et al., 2012b).*

One noteworthy issue that happens is the way t the number of scenarios considered grows exponentially with the number of time periods. This is evident in the sense that the final number of scenarios is extremely large once the number for each time period is high not minding the size of the time horizon. Thus, it becomes vital to produce a bargain solution so that without considering an unnecessary number of scenarios, neither extremely illustrative scenarios nor those liable to happen, will be eliminated since the issue considered thusly can't be managed. (Agustín et al., 2012b).

tion and the weighted probability of maximising the occurrence of scenarios"

[13]robust optimisation is "minimising the difference between the value of the proposed solution and the air traffic management solution value for each scenario (Agustín et al., 2009).



$$\Omega = \Omega_1 = \{1, 2, \ldots, 8\}; \quad \Omega_2 = \{1, 2, \ldots, 8\}$$
$$\mathcal{R} = \{1, \ldots, 34\}; \quad \mathcal{R}_2 = \{2\}; \quad \mathcal{R}_3 = \{3, 4, 5\}$$
$$\mathrm{a}(9) = 6; \quad t(9) = 5$$
$$\mathcal{V}_9 = \{1, 2, 3, 6, 9\}; \quad \mathcal{V}^9 = \{14, 19, 20, 27, 28\}$$

Figure 3.12: Illustration of a scenario tree (Agustín et al., 2012b).

Stochastic Models: Liu et al. (2008) as well as Buxi and Hansen (2011) developed techniques to determine probabilistic capacity profiles and scenario tree forecasts from historical data. Marron (2004) proposed a *column generation approach* for solving the ATFM problem with rerouting, and demonstrated it on a regional-scale example with approximately 150 flights whereas Gupta and Bertsimas (2011) studied robust and adaptive optimisation formulations of the ATFM problem to address capacity uncertainties

Balakrishnan and Chandran (2015) also considered the stochastic case of ATFMRP. They extended their approach to generate recourse strategies in the presence of uncertain capacity constraints. The solution provided for solving the problem is similar to the deterministic case only that it is no longer on a single path via a space-time network rather on a tree in a scenario space-time based network. The authors also used the *column generation approach* to solve the LP relaxation of the problem. Using dynamic programming algorithm, they solved a maximum weighted tree in a space-time-scenario network which is equivalent to solving the sub problem. The difference between their models and other models is based on the representation of the O-D routes by a standard node-link network which is unrestricted. Their model also considers almost all the control actions and was shown to be performing better computationally and adequate for real time implementation (Balakrishnan and Chandran, 2015).

Agustín et al. (2012b) extended Agustín et al. (2012a), the deterministic case of their ATFM model, to account for uncertainty as well as rerouting. They formulated it as a *multi-commodity network dynamic flow model taking into account uncertainty in the airports and air sectors capacity and the uncertainty on the flight demand by the airline companies* (Agustín et al., 2012b). The proposed model uses a *scenario generation scheme* based on storm scenarios to represent the *Deterministic Equivalent Model (DEM)* of the stochastic mixed 0-1 program with full recourse in such a way that the *non-anticipativity constraints (NAC) for the continuous and 0-1 variables in each scenario group of each period along the time horizon are implicitly satisfied in the compact representation of DEM* Agustín et al. (2012b). Their work falls under the risk neutral environment category. The model computational performance was shown to be very good. See Agustín et al. (2009) for a comprehensive synopsis of the main features of the proposed model in Agustín et al. (2012b), which they tagged as Stochastic Air Traffic Flow Management (SATFM).

Chang (2010) proposed a *two-stage stochastic IP model* to solve the ATFMP with emphasis on a given single sector. The author considers the following control actions: *ground delay, cancellation, and cruise speed for each flight on the ground in the first stage and air holding and diversion recourse actions for each flight in the air in the second stage.* Weather uncertainty is not left out in the model formulation. To overcome the intractability of the proposed model which is caused by the weather scenarios, the author proposed *a rolling horizon method to solve the the problem to near optimal which was justified using Lagrangian relaxation and sub-gradient method.* The model was later extended to a multi-stage stochastic programming methodology. Same rolling horizon method, Lagrangian relaxation as well as sub-gradient method were applied to the multi-stage stochastic programming and comparison was made between the two methodologies (Chang, 2010).

Ronald (2010) developed stochastic optimisation models that reduces air traffic delay and at the same time allows for efficient traffic flow management actions by investigating dynamics of air traffic delay in Uganda. This was achieved by investigating on *the relationship between airport utility and the interaction effects of daily probabilities of delay and airport inefficiency estimates.* The analysis was carried out using data instances of 4 years interval obtained from Entebbe International Airport in order to make valid conclusion. Several conclusions were drawn based on the analysis and computational experiments which will be very useful in the decision making and management of the air traffic flow as well as Entebbe Airport Traffic managers. To the best of the researcher's knowledge, this is the only work that used instances and domestic flight data from African Region to solve the ATFMP (Ronald, 2010).

Ichoua (2013) addressed the issue of ATFMP with *stochastic time-dependent sector capacities and presented a scenario-modeling framework* to formulate the problem. The solution framework proposed in the work was based on *Sample Average Approximation Method (SAA) and a Deterministic MIP Model* which includes almost all the control actions as in Bertsimas et al. (2011). The SAA model is solved using CPLEX which gives several alternative solutions to select from and an evaluation of these alternative solutions gives the best plan (Ichoua, 2013).

3.3.2.3 Heuristic Algorithms and Other Approaches for the ATFMP

Tian and Hu (2010) developed a *non-linear multi-objective and multi-constrained model which utilises a multi-genetic algorithm* to reduce flight delays and air space congestion since dependence on a single-objective models to address problems like re-routing of flights, departure slots and air traffic workloads cannot meet the demands of ATFM (Tian and Hu, 2010).

Kuhn and Raith (2010) reformulated the ATFMP as a formal multi-objective optimisation problem in order to minimise *inefficiency and inequity.* This they achieved by finding all the *pareto-optimal solutions, trading off efficiency and equity without having to select and parametrise a model of the cost of inequity.* The most important aspect of their work is the consideration of equity which is a major goal of ATFM activities, in their formulation. Until now, the issue of equity and fairness has not been fully addressed in theory, as such this gap is one of the primary reasons why results from previous studies have not been fully adopted into practice in air traffic control. The need to delve into these issues and incorporate them in model formulation vis a vis addressing ATFMP becomes dire.

Barnier and Brisset (2004) addressed the problem from a graph theoretical point of view using the concept of graph colouring. The authors designed *new route network that considers direct routes only* and vertically separate intersecting ones by allocating distinct flight levels, thus leading to a *graph colouring* problem. The concept of *constraint programming* was used in solving the problem and a *greedy algorithm* was developed for obtaining large cliques which guides the search strategy and at the same time used for posting global constraints (Barnier and Brisset, 2004). Optimal solutions were achieved in almost all the instances except for the large ones with an implementation using *FaCiLe, Functional Constraint Library,* while the corresponding number of flight levels could fit in the current airspace structure. This graph colouring technique has also been tested on various benchmarks, featuring good results on real-life instances, which systematically appear to contain large cliques (Barnier and Brisset, 2004).

Torres (2012) approach for solving the ATFMP was nature inspired, in particular, the swarm theory. The idea is to *convert pilots into goal seeking agents* that individually find local solutions to the optimisation problem and as a whole the collective action of agents creates emergent behaviour that naturally tends to converge on its own to a Pareto efficient state. Since air traffic is *a multi-swarm system*, they presented a *Self-Managed Air Traffic (Self-MAT)* concept which removes the sources of inefficiency and provides mechanisms so that optimality results as an emergent behaviour (Torres, 2012).

Multiple Airport Scheduler (MAS) is a hybridised heuristics algorithm developed for the Advanced Traffic Management System of the FAA with the goal of *maximising flows in the ATM network subject to airport and en route capacities, flight connectivity and fairness-to-user constraints* (Epstein et al., 1992). Bertsimas and Odoni (1997) pointed out that the quality of the resulting solution of MAS cannot be ascertained as information regarding the model is limited.

Research findings indicates that EUROCONTROL uses a heuristic algorithm, popularly known as *Computer Assisted Slot Allocation (CASA) algorithm*, to support its tactical level planning when for solving the ATFMP (Philipp and Gainche, 1994; Bertsimas and Odoni, 1997). The description of the algorithm is omitted herein as Philipp and Gainche (1994) have given a detailed description of the algorithm. The algorithm is characterised by its priority preference which is done on a *First Planned, First Served (FPFS) basis*, that is, giving priority to flights with the earliest departure times. Moreover, the algorithm assigns ground holding delay sequentially to the flights with earliest departure times when necessary (Bertsimas and Odoni, 1997; Philipp and Gainche, 1994).

The CASA algorithm has a very efficient computational performance in the sense that solutions are returned in about 30 seconds for problems considered to be large for an optimisation model. However, there are cases where the LP or MIP formulations perform better than the search techniques depending on the polyhedral structure and valid inequalities used in the formulation. Although the performance was shown to be good, the quality of the solutions returned for ATFM slot allocations as well as ground-holding assignments by the the same algorithm is yet to be confirmed (Bertsimas and Odoni, 1997; Philipp and Gainche, 1994). The algorithm solutions are updated automatically every few minutes with the aim of finding an improved slot allocation strategy as conditions change. Bertsimas and Odoni (1997) noted that "*the algorithm reserves a portion of the available capacity for short-haul flights and (or) for flights that may, for some reason, file a flight plan shortly before their intended departure time*" (Bertsimas and Odoni, 1997). The major reason behind this initiative is that *all available slots might be consumed by long-haul flights that file flight plans early and have early departure times in the absence of such practice* (Bertsimas and Odoni, 1997; Philipp and Gainche, 1994).

Computational results for selected ATFMP models.

In this section, we briefly discuss some computational results for MAGHP and ATFMP considering Bertsimas and Patterson (1998) models, identified as the best existing optimisation model and has been widely tested utilizing genuine data from both the US and the European networks.

For the MAGHP, the datasets for the ground holding problem instances were the same data sets used in Vranas et al. (1994). More precisely, the datasets consist of two (2) and six(6) airports with 500, 1000 and 3000 flights per airport respectively. The authors considered 15-minutes time interval over a 16-hour day and four distinct levels of connectivity, which is a marker of the proportions of continued flights to aggregate flight. They adjusted the datasets keeping in mind the end goal to suit the contrasts between their model and that of Vranas et al. (1994). All experiments were performed on a SUN SPARC workstation 10 model 41 using GAMS as the modeling tool and CPLEXMIP 2.1 as the solver. Table 3.1 and 3.2 belows shows the summarized results obtained by the authors using the above mentioned datasets. For each case considered, the results were obtained at the infeasibility border[14], the reported times are in CPU and the percentage of the aggregate flights with non integer solutions displayed in the % Nonint column. The results when compared with that of Vranas et al. (1994) and showed a great improvement in the integrality of solutions.

Table 3.1: Bertsimas and Patterson (1998) results at infeasibilty border for 1000 flights

$\lvert F\rvert$	$\lvert C\rvert/\lvert F\rvert$	Dep Capacity	Arr Capacity	Time	% Nonint
1000	0.20	32	15	262	0
1000	0.40	17	10	741	4
1000	0.60	20	14	359	0
1000	0.80	20	20	283	0

Table 3.2: Bertsimas and Patterson (1998) results at infeasibilty border for 3000 flights

$\lvert F\rvert$	$\lvert C\rvert/\lvert F\rvert$	Dep Capacity	Arr Capacity	Time	% Nonint
3000	0.20	20	20	5475	0
3000	0.40	20	20	4703	0
3000	0.60	20	20	5407	0
3000	0.80	20	20	9411	0

For the ATFMP, they performed different set of experiments using data sets and pseudo generated datasets obtained from the Official Airline Guide (OAG) flight schedules. The first set of experiments consisted of 200 and 1000 flights over a 24-hour time period divided into 5 and 15 minutes interval respectively. The problem was solved using the same configuration setting as the previous one. The resulting optimal solution was integral in most cases. The other set of experiments were two realistic sized datasets obtained straightforwardly from the Official Airline Guide (OAG) flight schedules, were provided by the FAA. The first comprised of 278 flights, 10 airports and 178 sectors, tested over a 7 hour time horizon with 5 minute interims. The second of these information sets comprised of 1,002 flights, 18 airports, and 305 sectors tried over an 8 hour time span with 5 minute interims. The flight-sector crossing times, airport and sector capacities, turnaround times were all given by the FAA. These information sets are equivalent to those problems being solved every day by the FAA. The first problem when implemented consists of 43,226 constraints and 18,733 variables. An ideal (optimal) solution for the LP

[14] The set of critical values for the arrival and departure capacities in units of time per time interval under which the problem becomes infeasible

relaxation was found in roughly 30 minutes on a SUN SPARC 20 workstation utilizing CPLEX 3.0 as the solver and GAMS 2.25 as the modelling language. Moreover, the resulting solution were totally integral which implies that there is no need to utilize any integer programming techniques. For the second and bigger information set comprising of 151,662 constraints and 69,497 variables, was solved to optimality in around 2 hours with the resulting completely integral. Comparable results, with basically the same model, were gotten by Vranas et al. (1994) for the European system. For an information set provided by EUROCONTROL, a problem instance including 2,293 flights and 25 sectors and with all costs equivalent to one, that is minimising the aggregate delay, was solved to optimality and the solution found in around one hour in a SUN 10 workstation. See Vranas et al. (1994), Bertsimas and Patterson (1998) and Bertsimas and Odoni (1997) for more details. It is worthy to note that in all the instances considered for the MAGHP and ATFMP, integral solutions were obtained from the LP relaxations which was not affected by the problem size or parameters used. The solutions were obtained within the shortest possible computational time.

3.4 Chapter Summary

The chapter focused mainly on the existing approaches, initiatives and methodologies that have been (and continue to be) used to address the ATFMP. The ATFMP was introduced and briefly discussed after which an insight was given on the initiatives and techniques currently used by CAMU in addressing the Air Traffic Flow Management within the South African airspace. Particular attention was given to the existing optimisation models for the ATFMP that focus more on airport capacity and airspace allocation. These were extensively reviewed because they depict what is obtainable in the air transportation system. The ground holding approach emerged to be more effective in situations where *congestion at the airports is most likely to occur and less effective when airspace sector congestion is also considered since the flow of air traffic between origin and destination airports has to be considered.* Moreover, the study revealed how both the airborne holding and ground delay can be reduced to lowest minimal once rerouting is included as one of the control actions for the ATFMP. However, there is need to incorporate other control actions when formulating the mathematical model as this will help in addressing real life situations. Furthermore, the study shows that much work has been done on this area of study using instances from American and European countries but little or no attention have been given to ATFM data instances from developing countries over the past decade. Thus, the need for researchers to shift their focus from American and European countries to what is obtainable in other developing and underdeveloped countries (as similar problems or even worse problem abound), and apply these methodologies to solve the ATFM problem and challenges associated with it. This will immensely help the concerned countries in decision making and consequently contribute positively to their economic growth and development.

A wide variety of optimisation algorithms/techniques (considering several objective functions, possible control actions, operational and capacity constraints) exist in literature for solving the ATFMP either for the deterministic or the stochastic case. However, since ATFMP has been shown to be *an NP hard problem*, the problem encountered is that *the run time needed to find an optimal solution increases exponentially with the size of the problem as well as the degree of connectivity.* Although extensive computational experiments tested on some national and regional instances which were used to validate existing models show that large-scale ATFMP can be solved to near-optimal solution within feasible computational times, these can still be reduced further if better optimisation techniques are applied as well as exploring other LP and MIP optimisation solvers. Current research has shown that application of hybrid-metaheuristic and metaheuristic optimisation methods on most complex real-world problems can produce optimal solutions within the shortest possible time. It would be a good option therefore to explore these options in solving the ATFMP (Bertsimas and Patterson, 1998; Chaimatanan, 2014; Agustín et al., 2012b).

A practical ATFM is required to maintain *a careful balance between equity and efficiency* Kuhn and Raith (2010). *Equity* in this sense ensures that the costs incurred as a result of ATFM activities do not dis-proportionally fall on certain airlines or flights. Although the different proposed mathematical models/optimisation techniques have been validated for solving ATFMP via its computational performance, they have not yet been fully adapted in practice given that the issue of equity is yet to be fully considered as a goal of ATFM activities. Although efforts have been made recently by several researchers with respect to this, in particular Barnhart et al. (2012); Bertsimas and Gupta (2011); Kuhn and Raith (2010), there is need to explore more on these issues and incorporate them in the model formulation. This will help to bridge the gap between research and practice.

Table 3.3 presents a summary of the instances, possible control actions and test cases considered by authorities in literature to solve the Air Traffic Flow Management Problem. A comparison of the different methodologies and test cases of different models reviewed in this paper relative to other existing ones in literature was also drawn. Although an overview of the computational performance of the reviewed works is presented in the table. The computational run times are not necessarily comparable as there are significant differences in the computing environments. They are only presented for context purposes.

Table 3.4 is a summary table of the basic assumptions with respect to the different approaches for solving the Air Traffic Flow Management Problem. The computational complexity for the basic models with respect to each approach were also reported. Nonetheless, the complexity of algorithms for solving ATFMP varies depending on the formulation and solution approach adopted by the decision makers.

Table 3.3: Summary of ATFMP instances and control actions for selected references

References	Instances	Control Actions	Datasets	Horizon/Disc	Run time
Helme (1992)	Deterministic	Ground Holding and Air borne Holding			
Bertsimas and Patterson (1998)	Deterministic	Ground Holding and Air borne Holding	1002 flights, 18 airports, 305 sectors	8hours/5min	8+ hours
Bertsimas and Patterson (2000)	Deterministic	Ground Holding, Air borne Holding and Restricted Rerouting	71 flights, 4 airports, 42 sectors	8hours/ 5mins	4 mins
Lulli and Odoni (2007)	Deterministic	Ground Holding and Air borne Holding	368 flight arrivals, MXP airport, different sector cases	18hours/15 mins	
Bertsimas et al. (2008)	Deterministic	Ground Holding, Air borne Holding, Restricted Rerouting and Speed Control	6745 flights, 30 airports, 145 sectors	8hours/5 mins	10 mins
Bertsimas et al. (2011)	Deterministic	Ground Holding, Air borne Holding, Restricted Rerouting, Miles-in-trails and Speed Control	6745/3000 flights, 30/20 airports, 145/113 sectors	8hours/5min	10 mins
Marron (2004)	Stochastic	Ground Holding and Air borne Holding	148 flights, 40 sectors, 3 scenarios	not spec/5min	112 mins
Mukherjee and Hansen (2007)	Stochastic (dynamic)	Ground Holding and Air borne Holding			
Chang (2010)	Stochastic (dynamic)	Ground Holding, Air borne Holding, Diversion, Cruise Speed and Flight Cancellation			
Agustín et al. (2012a,b)	Deterministic & Stochastic	Ground Holding, Air borne Holding, Restricted Rerouting, Flight Continuation, Flight Cancellation, Speed Control etc.	Four testbeds including the one used by Bertsimas et al. (2011). See Agustín et al. (2012a,b) for details.		
Balakrishnan and Chandran (2015)	Deterministic & Stochastic	Ground Holding, Air borne Holding, Unrestricted Rerouting, Flight Cancellation and Speed Control			

Table 3.4: Summary of basic assumptions and computational complexity ATFMP

Approaches	Type	Selected References	Basic Assumptions	Complexity
Ground Holding Approaches	SAGHP	Odoni (1987), Terrab and Odoni (1993), Richetta and Odoni (1993), Hoffman et al. (1999), Richetta and Odoni (1994)	Assumptions: A1 - A4; B1 - B6	Polynomial time complexity
Ground Holding Approaches	MAGHP	Bertsimas and Patterson (1998), Vranas et al. (1994), Andreatta and Brunetta (1998),Andreatta and Tidona (1994),	Assumptions: C1 - C7. En route airspace capacity constraint is not considered.	NP Hard, Polynomial time complexity
Air Traffic Flow Management Approaches	BATFMP	Bertsimas and Patterson (1998), Bertsimas et al. (2011),Helme (1992), Lindsay et al. (1993), Lulli and Odoni (2007), Bertsimas and Gupta (2011)	Assumptions: D1 - D5. Rerouting, speed control are not considered as a control option. The flights route are predetermined	NP Hard
Air Traffic Flow Management Approaches	ATFMRP	Bertsimas et al. (2008), Balakrishnan and Chandran (2015), Bertsimas and Patterson (1998), Bertsimas et al. (2011), Bertsimas and Patterson (2000), Agustín et al. (2012a)	Assumptions: D1, D4, D5. Rerouting and flight speed are considered as a control option. The flights route are not predetermined. Routes can be represented as an acyclic graph or collection of subgraphs	NP Hard
Air Traffic Flow Management Approaches	SATFMP	Balakrishnan and Chandran (2015), Ronald (2010), Chang (2010), Liu et al. (2008), Agustín et al. (2012b),?, Marron (2004),Ichoua (2013), Torres (2012),?, ?	Same assumptions as ATFMRP but on probabilistic environment instead of deterministic. The assumption includes generic uncertainty in the parameters, represents uncertainty by set of possible scenarios and decisions are made at each point in time.	NP Hard, Polynomial time complexity and exponential time complexity

Chapter 4

Deterministic Mixed Integer Programming Models for the ATFMP

This chapter provides insight into the mathematical formulation of the proposed integer programming models for reducing air traffic delays and congestion problems. The models presented in Bertsimas and Patterson (1998) and Bertsimas et al. (2011) and the proposed approach for including rerouting in the basic ATFMP model presented in Bertsimas and Patterson (1998) formed the basis of the IP formulation described here because of its strong formulation and computational performance when handling large instances of data. Key ideas and basic assumptions from the underlying model that are relevant to the mathematical formulation of the ATFMRP are presented in Section 4.1. The proposed formulations are discussed in Section 4.2 based on the proposed sector and path approach.

4.1 Mathematical Problem Formulation

Bertsimas and Patterson (1998) formulated the basic ATFMP mathematically (see equation 3.3.5 − 3.3.15) and proposed two main approaches for incorporating rerouting options into the basic TFMP model, namely sector and path approaches. The decision variable of their model and the proposed approach provides direction for the model presented here. The goal/objective is to come up with a strong formulation using a better strategy that reduces airspace and airport congestion while satisfying the capacity constraints as well as ensuring safe and efficient flow of aircraft through the airspace.

The Starting Point

The basic model presented in Bertsimas and Patterson (1998) was described briefly in the previous chapter (see equation 3.3.5 − 3.3.15) and will not be repeated. Key ideas from the basic model and basic assumptions, relevant to the present work are included in this section to serve as a guide in understanding the proposed model. Moreover, the basic concepts described in Section 2.2 form part of the underlying principles of the model formulation.

Basic Assumptions

1. The airspace is divided into sectors and can be represented as *a graph with the nodes and edges representing the sectors and the neighbourhood* respectively (Bertsimas

and Patterson, 1998; Bertsimas et al., 2011; Delahaye and Puechmorel, 2013).

2. Flights are expected to fly through *adjacent sectors* en route to its arrival airport (Bertsimas and Patterson, 1998).

3. There is restriction on *the number of aircraft to cross a sector* at each time period[1].

4. The origin to destination (O-D) can be represented as *a sequence of sectors* flown by a flight. The O-D route is predetermined if rerouting is not included. To incorporate rerouting option, each *O-D route can be represented by a digraph where the set of nodes of the digraphs represents sectors and airports that is the set of capacitated elements of the airspace* (Agustín et al., 2009; Bertsimas et al., 2011). Moreover, the idea of Partially-Ordered Sets (POSETS) helps in the description of the O-D route of a flight especially when using the sector approach to rerouting. See Bertsimas et al. (2011) for more details.

5. The *time horizon, T , is fixed and discretised into equal-sized, contiguous time periods, $t = 1, \cdots T$* (Bertsimas and Patterson, 1998).

6. The capacities, departure time and flight time are *deterministic in nature.*

7. *Flight continuation and flight cancellation are considered.*

8. There is *a limit on the amount of ground holding and airborne delay to be assigned to each flight.*

9. There is *maximum accepted duration of flight for each flight.* Thus, flights that exceed the maximum duration of flights are penalised.

Recall from Section 3.3.2, that the input data sets, parameters and variables have been defined. The flight sector variable w^j_{ft} are defined as *being 1 if flight f arrives at sector j by time t* expressed as

$$w^j_{ft} = \begin{cases} 1, & \text{if flight } f \text{ arrives at sector } j \textbf{ by } \text{time } t, \\ 0, & \text{otherwise} \end{cases} \tag{4.1.1}$$

which is a variant of the standard flight-sector variable defined as

$$u^j_{ft} = \begin{cases} 1, & \text{if flight } f \text{ arrives at sector } j \textbf{ at } \text{time } t, \\ 0, & \text{otherwise} \end{cases} \tag{4.1.2}$$

P_f have been previously defined to be the sectors to be flown by a flight, that is, the predefined route of the flight. Also, T^j_f is the defined to set of feasible time periods for

[1]This restriction is referred to as **en route sector capacities** (Bertsimas and Patterson, 1998).

a flight to be in a particular sector. Thus, the decision variable w^j_{ft} are defined only for those sectors in P_f and those times within T^j_f.

Moreover, Bertsimas and Patterson (1998) noted that "*one variable per flight-sector pair can be eliminated from the formulation by setting* $w^j_{f,\overline{T}_{fj}} = 1$ *as a parameter* where $\overline{T}_{fj}$ is the last time period in the set T_{fj} since flights has to arrive at sector j by the last period" (Bertsimas and Patterson, 1998, pg 409). The illustrative example in 4.1.1 will help in understanding of the model.

4.1.1 Example. *Fig 4.1 is an illustrative diagram showing two flights travelling through a set of sectors.*

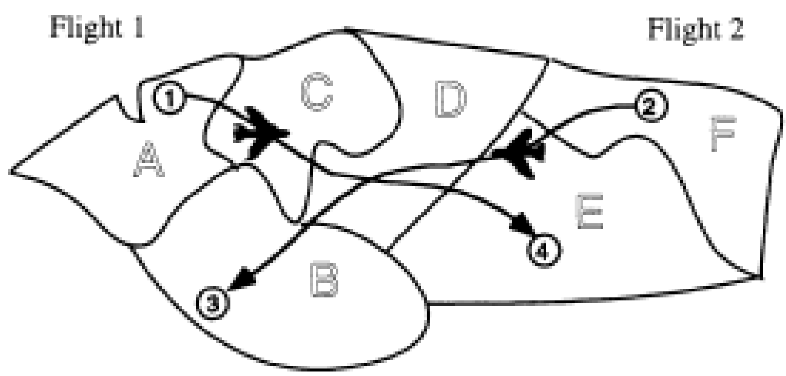


Figure 4.1: Illustrative example of flight routes (Bertsimas and Patterson, 1998).

The associated data for each of the flight is given as $P_1 = (1, A, C, D, E, 4)$ *and* $P_2 = (2, F, E, D, B, 3)$*. If "the current position of the aircraft is assumed to occur at time* t*, then the variables for these flights at time* t*" will be:* $w^1_{1,t} = 1, w^A_{1,t} = 1, w^C_{1,t} = 1, w^D_{1,t} = 0,$ $w^E_{1,t} = 0, w^4_{1,t} = 0,$ *and* $w^2_{2,t} = 1, w^F_{2,t} = 1, w^E_{2,t} = 1, w^D_{2,t} = 0, w^B_{2,t} = 0, w^3_{2,t} = 0$ *(Bertsimas and Patterson, 1998).*

For a clear understanding of the underlying models, Bertsimas and Patterson (1998); Bertsimas et al. (2008, 2011), it is worthy to note some of the factors that contributes to the excellent computational performance of the models. Thereby distinguishing them from other models found in literature. They include:

1. Strong formulations of the models.
2. Facet-defining constraints and introduction of valid inequalities to strengthen the polyhedral structures.
3. The definition with "**by**" rather than "**at**" in the decision variables.
4. The transformation from u^j_{ft} to w^j_{ft} that is $u^j_{ft} = w^j_{ft} - w^j_{f,t-1}$ and $w^j_{ft} = \sum_{t' \leq t} u^j_{ft'}$, which are used to express several quantities of interest.

5. The solutions from the LP relaxations are completely integral solutions and are obtained within the shortest possible time.

The concepts described in Section 2.2, especially Theorem 2.2.18, explains the underlying principle behind the *occurrence of integral solutions* when solving ATFMP using strong formulation. See Wolsey and Nemhauser (1988); Bertsimas and Patterson (1998); Bertsimas et al. (2008, 2011) for more details.

The Objective Function

The global objective function, depending on the goal of the decision makers, can include different terms which can be joined together with suitable weights to form an objective function to be optimised. In as much as the objective function comprises of different terms to be minimised, the main goal as pointed out in Bertsimas et al. (2011) is to *minimise a function of total delay costs that is a combination of airborne and grounding holding delay costs* possessing the following properties

- It is expected that *airborne delays assigned to each flights should be more costly per unit of time than the ground delay* (Bertsimas et al., 2011)
- Moreover, delays should be assigned to flights in a *fairly* manner

Based on the above properties, the expression used in Bertsimas et al. (2008) of the form $\alpha AD + GD$ for objective function was adapted for this work where AD and GD represents airborne and ground delay respectively with equivalence factor, $\alpha > 1$, since *air-borne delay is assumed to be more costly than ground-holding delay.* Thereby modifying the objective function of the basic model.

The following approximations can be done based on the choice of expression $\alpha AD + GD$

$$\alpha AD + GD = \alpha AD + \alpha GD - \alpha GD + GD \tag{4.1.3}$$

or

$$\alpha AD + GD = \alpha AD - AD + AD + GD \tag{4.1.4}$$

As a result, the expression for the objective function can be rewritten as

$$\alpha TD - (\alpha - 1)GD \text{ or } TD + (\alpha - 1)AD \tag{4.1.5}$$

where TD is the total delay given by

$$TD = AD + GD \tag{4.1.6}$$

This implies that the objective function is either *the difference between the total delay cost and the cost reduction obtained by ground holding* or *the sum of total delay cost assigned to a flight and an additional cost incurred by airborne delay* depending on the choice of approximation (Bertsimas et al., 2008; Lulli and Odoni, 2007; Bertsimas and Gupta, 2011). For the purpose of this work, the choice of approximation is the expression given in equation 4.1.3.

To ensure *equity* among flights, that is *assigning moderate assignment of total delays between two flights* rather than assigning larger amount of delay to one as compared to the other flight, we adapt the same objective function cost coefficient, of the form $(t - a_f)^{1+\epsilon}$ for each flight f and for each time period t with ϵ close to zero that are *super-linear function of the tardiness of a flight* as shown in Bertsimas et al. (2008). The following example elaborates more on why the super-linear cost coefficients are priviledged.

4.1.2 Example. *Suppose, two flights are to be assigned two units of delay, then the following assignments is what is likely to be obtainable using an objective function with linear coefficients*

- *1 unit of delay to both flights and*
- *2 units of delay to one flight and 0 to the other*

If the latter is selected, then the delay is not fairly assigned to the flights. In contrast, a super-linear cost coefficient (with $\epsilon = 0.001$ *for example) will assign 1 unit to both flights since* $1^{1.001} + 1^{1.001} < 2^{1.001} + 0$ *(Bertsimas and Gupta, 2011).*

Based on the first choice of (4.1.5), the following cost coefficients are defined for each flight f and period t. Thus, the flight cost C^f_{ts}, that is, the cost of flight to arrive to a sector in it's path at time period t, can accommodate the super-linear function as defined below:

- By letting $C^f_{tD_f} := C^f_{td}(t)$ to denote *the total cost of delaying flight f (ground holding, airborne and rerouting) for $(t - a_f)$ units of time periods delay*, where t and a_f represents the actual and scheduled arrival time of flight. $C^f_{td}(t)$ can be expressed as

$$C^f_{td}(t) \equiv \alpha(t - a_f)^{1+\epsilon} \tag{4.1.7}$$

  with α and ϵ denoting the equivalence factor and super cost coefficient respectively.

- Similarly, letting $C^f_{tO_f} := C^f_{gr}(t)$ to denote *the cost reduction obtained by holding flight f on the ground for $(t - d_f)$ units of time periods delay*, where t and d_f represents the actual and schedule departure time of departure respectively. The expression for $C^f_{gr}(t)$ is given as

$$C^f_{gr}(t) \equiv (\alpha - 1)(t - d_f)^{1+\epsilon} \tag{4.1.8}$$

A combination of both costs in (4.1.7) and (4.1.8) gives *the super-linear cost of total delay minus ground holding reduction cost* which forms part of the objective function to be optimised. The total delay cost defined in (4.1.7) can be used in place of the ground delay and air borne delay costs (c_f^g and c_f^a) defined in Section 3.3.2 for the BATFMP or together with any other fixed or variable costs for delaying a flight or holding the flight on the ground. Moreover, by choosing parameters ϵ_2, ϵ_1 close to zero to be coefficients for airborne and ground holding delay respectively in such a way that ($\epsilon_2 > \epsilon_1$), the flight cost C_{ts}^f can accommodate the super-linear function in Bertsimas et al. (2011) as illustrated below:

$$C_{td}^f(t) \equiv \alpha(t - a_f)^{1+\epsilon} = (t - a_f)^{1+\epsilon_2} \tag{4.1.9}$$

and

$$C_{gr}^f(t) \equiv (\alpha - 1)(t - d_f)^{1+\epsilon} = (t - a_f)^{1+\epsilon_2} - (t - d_f)^{1+\epsilon_1} \tag{4.1.10}$$

Thus, the BATFMP can be modified with respect to the above approximation using both costs in the objective function.

Modification of the BATFMP

In view of the above, the *delay cost function for each flight f at each time period t* takes the form:

$$\sum_{t \in T_f^k : k = P(f,N_f)} \alpha(t - a_f)^{1+\epsilon}(w_{f,t}^k - w_{f,t-1}^k) - \sum_{t \in T_f^k : k = P(f,1)} (\alpha - 1)(t - d_f)^{1+\epsilon}(w_{f,t}^k - w_{f,t-1}^k) \tag{4.1.11}$$

Thus, the modified objective function becomes

$$Min \sum_{f \in F} \Big[\sum_{t \in T_f^k : k = P(f,N_f)} C_{td}^f(t)(w_{f,t}^k - w_{f,t-1}^k) - \sum_{t \in T_f^k : k = P(f,1)} C_{gr}^f(t)(w_{f,t}^k - w_{f,t-1}^k) \Big] \tag{4.1.12}$$

The set of feasible time window for a flight to arrive at a particular sector in Bertsimas and Patterson (1998) was not explicitly defined. To explicitly define these sets, additional parameters and new constraints were to incorporated into the BATFMP and proposed formulation. The parameters include: *the maximum units of delay for a flight to be held on the ground or in the air, the maximum acceptable duration of flight, scheduled time for a flight to arrive at a particular sector from its origin airport, the scheduled time for flight to cross a sector while the additional constraints ensures that flight does not exceed the maximum acceptable duration of flight.* For the purpose of this work, the following sets are explicitly defined:

1. The set of feasible time window for a flight to depart from its origin airport is defined as

$$T_f^k = \{t \in T : d_f \leq t \leq min(d_f + G_f, T)\} \ \forall f \in F : k = P(f, 1) \qquad (4.1.13)$$

where f, F, $k = P(f, 1)$, t, d_f, T takes their usual meaning and "*the maximum units of delay that can be assigned to a flight on the ground*" is denoted by G_f.

2. The set of feasible time window for a flight to enter its destination is defined as

$$T_f^k = \{t \in T : a_f \leq t \leq min(a_f + G_f + A_f, T)\} \ \forall f \in F : k = P(f, N_f) \qquad (4.1.14)$$

where f, F, $k = P(f, N_f)$, t, a_f, T takes their usual meaning and G_f, A_f represents the "*maximum units of delay that can be assigned to a flight on the ground and in the air*" respectively.

3. The scheduled time for a flight to arrive at a particular sector in it's flight path can be expressed as

$$\underline{T}_f^j = d_f + T_{o,j} \equiv d_f + \sum_{i=P(f,j'):j'<j} l_{fi} \ \forall f \in F, j \in P_f \qquad (4.1.15)$$

where $T_{o,j}$ is the defined time for a flight to reach a sector from its origin airport, $\underline{T}_f^j$, the last time period in T_f^j, $l_{f,j}$, the minimal transit time for a flight to traverse from one sector to another.

4. The set of feasible time window for a flight to arrive at a particular sector on its flight path is defined as

$$T_f^k = \{t \in T : \underline{T}_f^j \leq t \leq min(\underline{T}_f^j + G_f + A_f, T)\} \qquad (4.1.16)$$

$\forall f \in F, j = P(f, i) : 1 < P(f, i) < N_f$ where f, F, $P(f, i)$, t, T, G_f, A_f denotes its usual meaning.

4.2 Insight on the proposed approach for the ATFMRP

The sector and path based approach for ATFMRP as illustrated in Bertsimas and Patterson (1998) with the proposed complete formulation for each approach are presented below.

4.2.1 Sector-Based Approach

This approach *decides at each sector in a flight's route which sector to enter next* (Bertsimas and Patterson, 1998). To implement this approach, new sets are defined namely N_{fj} and $P_{f,j}$. N_{fj} is the set of sectors that a flight f can enter immediate after leaving a particular sector, say sector j, while P_{fj} is the set of sectors that flight f can enter immediate before entering sector j (Bertsimas and Patterson, 1998). P_f was previously defined as the predefined route for each flight including the departure and arrival airports but in this approach, P_f is now defined as the *set of sectors to be flown that can be flown by flight f, including the origin and destination airport* denoted now as S_f. See Example 4.2.1 for illustration. An additional index, j', that keeps track of the preceding sector is introduced in the decision variables. Thus, *the new decision variables* for incorporating rerouting based on the sector approach as presented in Bertsimas and Patterson (1998) is given below

$$w_{ft}^{jj'} = \begin{cases} 1, & \text{if "flight } f \text{ arrives at sector } j' \text{ from sector } j \textbf{ by } \text{time } t\text{"}, \\ 0, & \text{otherwise} \end{cases} \tag{4.2.1}$$

Moreover, the decision variable that was previously defined for the BATFMP can be expressed in terms of this new decision variable as shown in (4.2.2) below

$$w_{ft}^{j} = \sum_{j' \in N_{fj}} w_{ft}^{jj'} \tag{4.2.2}$$

4.2.1 Example. *Figure 4.2 is an illustration of "possible routes from an origin airport to a destination airport"(Bertsimas et al., 2011). Suppose we assume that a flight f' is departing from origin, O, then there are three (3) possible routes that the flight can follow to reach D, its destination. Based on the sector approach, the associated data for the flight*

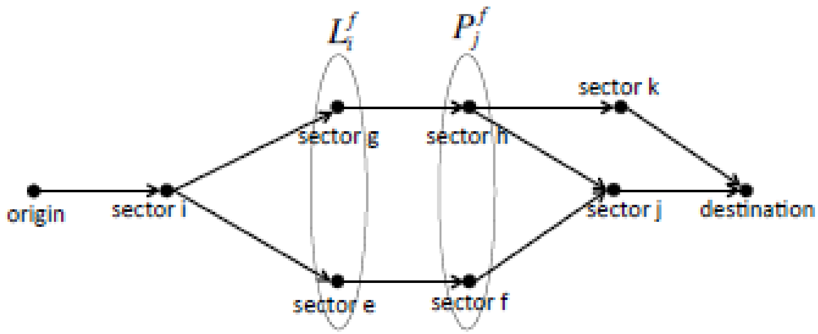


Figure 4.2: Illustration of possible routes from O-D airport (Bertsimas et al., 2008, 2011).

is given as $S_{f'} = \{O, i, g, h, e, f, k, j, D\}$. $N_{f'O} = \{i\}, N_{f'i} = \{g, e, \}, N_{f'g} = \{h\}, N_{f'h} = \{k, j\}, N_{f'e} = \{f\}, N_{f'f} = \{j\}, N_{f'k} = \{D\}, N_{f'j} = \{D\}$. $P_{f'i} = \{O\}, P_{f'g} = \{i\}, P_{f'h} =$

$\{g\}, P_{f'e} = \{i\}, P_{f'f} = \{e\}, P_{f'k} = \{h\}, P_{f'j} = \{h, f\}, P_{f'D} = \{k, j\}$. *There is nothing like* $P_{f'O}$ *and* $N_{f'D}$ *since they represent the departure and arrival airport.*

In addition to the above description, the departure and arrival airports remain unchanged with respect to this new decision variable for all flights. Thus, if we denote the departure and arrival airports by O_f and D_f respectively, then $w_{ft}^{O_f,O_f} = w_{ft}^{O_f}$ since it is expected that flights departs from the origin airport and arrives at the sector containing the airport by time t. Likewise, $\sum_{j \in P_{fj}} w_{ft}^{j,D_f} = w_{ft}^{D_f}$ since a flight is expected to arrive at it's destination airport from one of the preceding sectors by time t. These variables in most cases, takes on the value 1 except if the flight is cancelled. In other words, the flight would not have departed from its origin airport, hence will not arrive to its destination.

Model definition and Notation

The formulation of the model requires definition of the following notation for the input data sets and parameters to be used. Some of the notations used for the underlying model were still maintained for the proposed formulation. The input data sets and parameters for the model formulation include:

Input Sets: :=

$$
\begin{aligned}
F &:= \{1, \cdots, \mathcal{F}\}, \quad \textit{set of flights}, \\
K_d &:= \{1, \cdots, \mathcal{K}_d\}, \ \textit{set of departing airports} \\
K_a &:= \{1, \cdots, \mathcal{K}_a\}, \ \textit{set of arrival airports} \\
K &:= \{1, \cdots, \mathcal{K}\}, \ \textit{set of airports}, \ K = K_d \cup K_a. \\
T &:= \{1, \cdots, \mathcal{T}\}, \quad \textit{set of time periods}, \\
S &:= \{1, \cdots, \mathcal{S}\}, \quad \textit{set of sectors}, \\
C &:= \{(f', f) : f' \text{ is continued by } f\}, \textit{set of continued flights} \\
S_f &:= \textit{set of sectors that can be flown by flight including O-D airports} \\
N_{fj} &:= \textit{set of sectors that flight } f \textit{ can enter immediately after exiting sector } j \\
P_{fj} &:= \textit{set of sectors that flight } f \textit{ can enter immediately before entering } j \\
&\quad\ S_f \subseteq S; \ \forall j \in S_f, \quad P_{fj} \subset S_f \text{ and } N_{fj} \subset S_f \\
\underline{T}_{fj} &\equiv \textit{first time period in } T_j^f \\
\overline{T}_{fj} &\equiv \textit{last time period in } T_j^f \\
T_j^f &:= [\underline{T}_{fj}, \overline{T}_{fj}] \equiv \textit{set of feasible times for flight to be in sector } j
\end{aligned}
$$

The last three sets are calculated sets in preprocessing. The definition of the sets follows closely to that already defined in the previous section.

Input Parameters: :=

O_f := *origin (departure) airport of flight f*

D_f := *destination (arrival) airport of flight f*

$D_k(t)$:= *departure capacity of airport k at time t*

$A_k(t)$:= *arrival capacity of airport k at time t*

$S_j(t)$:= *capacity of sector j at time t*

d_f := *scheduled departure time of flight f*

a_f := *scheduled arrival time of flight f*

s_f := *turnaround time for an airplane after flight f*

l_{fj} := *minimum number of time units that flight must spend in j*

G_f := *maximum ground holding time units delay for flight f*

A_f := *maximum air borne time units delay for flight f*

M_f := *maximum allowed number of time units delay for flight f to arrive to its destination without penalisation.*

max_f := *maximum acceptable duration of flight f*, $max_f = a_f - d_f + A_f$

C_{cf} := *cancellation cost of flight f*

C^f_{ts} := *cost of flight to arrive to a sector in it's path at time period t*, $\forall f \in F,\ t \in T^f_j,\ j \in S_f : j \neq O_f$. $C^f_{ts} = 0$ *if arrival is on schedule.*

$schd_{fj}$:= *scheduled travel time for flight to arrive at j* $schd_{fO_f} = schd_{fD_f} = 0.$

$schd_{min}$:= *minimum travel time allowed for flight f to arrive a sector*

$schd_{max}$:= *maximum travel time allowed for flight f to arrive a sector*

$$schd_{min} \leq schd_{fj} \leq schd_{max},$$

$$schd_{min(O_f)} = schd_{fO_f} = schd_{max(O_f)} = 0$$

$$schd_{min(D_f)} = schd_{fD_f} = schd_{max(D_f)} = 0$$

By letting $C^f_{tD_f}$ and $C^f_{tO_f}$ to be the delay cost of flight to arrive to its destination and ground holding delay cost respectively, $C^f_{tO_f} = 0$ for $t \leq d_f$ and $C^f_{tD_f} = 0$ for $t \leq a_f$.

The Objective Function

With respect to the description of the objective function presented in the previous section, the objective function to be optimised is presented below based on the new decision variables. As pointed out earlier, these objective functions can be combined with appropriate weights to form a single objective function. See Agustín et al. (2012a); Bertsimas et al. (2011) for more details.

- *The difference between the total cost of delaying the flights f for time $(t - a_f)$ time units and the cost reduction obtained by holding flight for $(t - d_f)$ time units on the ground.*

$$\sum_{f\in F}\Big[\sum_{j\in P_{fj}}\sum_{t\in T^f_{D_f}} C^f_{td}\Big(w^{j,D_f}_{f,t} - w^{j,D_f}_{f,t-1}\Big) - \sum_{t\in T^f_{O_f}} C^f_{gr}\big(w^{O_f}_{f,t} - w^{O_f}_{f,t-1}\big)\Big] \tag{4.2.3}$$

- *The total cancellation cost.*

$$\sum_{f\in F}\sum_{t\in T^f_{O_f}} C_{cf}\Big(1 - w^{O_f}_{f,t}\Big), \tag{4.2.4}$$

  where $\sum_{f\in F}\sum_{t\in T^f_{O_f}}\Big(1 - w^{O_f}_{f,t}\Big)$ represents the number of canceled flights.

- *The number of flights exceeding the maximum allowed units of delay for arrival to their destination without penalisation*

$$\sum_{f\in F}\Big[\sum_{j\in P_{fj}}\sum_{t\in T^f_{D_f}}\Big(w^{j,D_f}_{f,t} - w^{j,D_f}_{f,a_f+M_f}\Big)\Big] \tag{4.2.5}$$

- *The cost of arriving at each sector, j, in the flight routes at each time period, t*

$$\sum_{f\in F}\Big[\sum_{j'\in p_{fJ}}\sum_{t\in T^f_j} C_{st}\Big(w^{j',J}_{f,t} - w^{j',j}_{f,t-1}\Big)\Big] \tag{4.2.6}$$

4.2.2 Path-Based Approach

This approach decides from the onset *which of the routes a flight takes from available options of routes to reach its destination.* To use this approach, the *set of possible routes that a flight may fly* needed to be defined. Let the set be denoted by R_f. R_f contains at least one route. For the BATFMP that was initially formulated, $R_f = P_f$ since it is a predefined route. The new decision variables for incorporating re-routing based on the path approach as illustrated in Bertsimas and Patterson (1998) is presented below. An additional index was included in the variable to show the route that the flight took to its destination.

$$w_{ft}^{jr} = \begin{cases} 1, & \text{if } \textit{flight } f \textit{ arrives at sector } j \textbf{ by } \textit{time } t \text{ along route } r, \\ 0, & \text{otherwise} \end{cases} \tag{4.2.7}$$

The basic decision variables defined for the BATFMP can also be expressed in terms of these current decision variables as shown in (4.2.8) below

$$w_{ft}^{j} = \sum_{r \in Q_f} w_{ft}^{jr} \tag{4.2.8}$$

Similar to the sector based approach, the departure and arrival airports remains unchanged with respect to this new decision variable for all flights. Thus, $w_{ft}^{O_f,r} = w_{ft}^{O_f}$ since it is expected that flights departs from an origin airport along a particular route r by time t. Likewise, $w_{ft}^{D_f,r} = w_{ft}^{D_f}$ since a flight is expected to reach it's destination airport along a preferred route by time t. As with the sector based approach, these variables takes on the value 1 except if the flight is cancelled.

4.2.2.1 Model Definition and Notation

The formulation of the model requires definition of the following additional notations for the input data sets and parameters to be used. The rest of the notations defined for the sector based approach except for S_f, N_{fj}, P_{fj}, C_{ts}^f remains valid for the path-based approach.

$$\begin{aligned}
R_f &:= \{r_{1f}, \cdots, r_{nf}\}, \textit{set of possible routes that flight } f \textit{ may choose} \\
r_{nf} &:= \{r(f,i) : 1 \le i \le n_{sf}\}, \\
n_{sf} &:= \textit{number of sectors in flight } f\textit{'s path} \\
r(f,i) &:= \textit{the } i^{th} \textit{ sector in flight } f\textit{'s path, where } r(f,1) = O_f \textit{ and } r(f,n_{sf}) = D_f \\
C_{st}^f &:= \textit{cost of flight to arrive to a sector in it's path at time period } t, \\
& \quad \forall f \in F,\ t \in T_j^f,\ j \in r_{nf} : j \neq r(f,1)\text{''}.\ C_{ts}^f = 0 \ \text{ if the arrival is on schedule} \\
C_{ar} &:= \textit{cost of using an alternative route different from the scheduled route}
\end{aligned}$$

As with the sector-based approach, $C_{tO_f}^f = 0$ for $t \le d_f$ and $C_{tD_f}^f = 0$ for $t \le a_f$ on the assumption that $C^f_{tD_f}$ and $C^f_{tO_f}$ represent the delay cost of flight to arrive to its destination and ground holding delay cost respectively.

The global objective function still remains the same as that of the sector-based approach. All the costs previously defined in equations (4.2.3 to 4.2.5) can still be considered as part of the objective function except for changes in the decision variables as shown below. Moreover, cost of using an alternative route can also be included in the objective function to be minimised.

- *The difference between the total cost of delaying the flights f for time $(t - a_f)$ time units and the cost reduction obtained by holding flight for $(t - d_f)$ time units on the ground*

$$\sum_{f \in F} \Big[\sum_{t \in T^f_{D_f}} C^f_{td} \Big(w^{D_f, r}_{f,t} - w^{D_f, r}_{f,t-1} \Big) - \sum_{t \in T^f_{O_f}} C^f_{gr} \big(w^{O_f, r}_{f,t} - w^{O_f, r}_{f,t-1} \big) \Big] \tag{4.2.9}$$

- *The total cancellation cost.*

$$\sum_{f \in F} \sum_{t \in T^f_{O_f}} C_{cf} \Big(1 - w^{O_f, r}_{f,t} \Big), \tag{4.2.10}$$

where $\sum\limits_{f \in F} \sum\limits_{t \in T^f_{O_f}} \Big(1 - w^{O_f, r}_{f,t} \Big)$ represents the number of cancelled flights.

- *The number of flights exceeding the maximum allowed units of delay for arrival to their destination without penalisation*

$$\sum_{f \in F} \Big(w^{D_f, r}_{f,t} - w^{D_f, r}_{f, a_f + M_f} \Big) \tag{4.2.11}$$

- *The cost of arriving at each sector, j, in the flight routes at each time period, t*

$$\sum_{f \in F} \Big[\sum_{j \in r_{nf}} \sum_{t \in T^f_j} C_{st} \Big(w^{j,r}_{f,t} - w^{j,r}_{f,t-1} \Big) \Big] \tag{4.2.12}$$

- *The cost of using alternative flight routes.*

$$\sum_{f \in F} \sum_{r \in R_f} C_{ar} w^{j,r}_{f,t} \tag{4.2.13}$$

Constraints

The constraints for the model formulation are classified into different categories: These are capacity, operational or connectivity and delay constraints. The bounds of the variables are also considered as part of constraints for the model formulation. The capacity constraints ensure that the capacity of the system resources are not exceeded while the operational constraints ensures the smooth operation of flight while en route to its destination. The delay constraints ensure that the flight does not exceed the maximum allowed time.

4.3 Mixed $0-1$ Integer Programming Models for ATFMRP

4.3.1 IP model for the ATFMRP (Sector-Based Approach)

The complete IP formulation based on the sector approach, a variant of the basic ATFMP model, for reducing air traffic delay and congestion in en route airspace is presented below. The interpretation of the constraints follow closely from the ones described for the underlying model. The constraints interpretation for both the main and alternative formulation were given simultaneously.

$$\text{Minimise} \sum_{f\in F}\Big[\sum_{j\in P_{fj}}\sum_{t\in T^f_{D_f}} C^f_{td}\Big(w^{j,D_f}_{f,t}-w^{j,D_f}_{f,t-1}\Big)-\sum_{t\in T^f_{O_f}} C^f_{gr}\big(w^{O_f,O_f}_{f,t}-w^{O_f,O_f}_{f,t-1}\big)\Big] \quad (4.3.1)$$

Subject to

$$\sum_{f\in F:O_f=k}(w^{k,k}_{f,t}-w^{k,k}_{f,t-1})\leq D_k(t)\ \ \forall\ k\in K, t\in T \quad (4.3.2)$$

$$\sum_{f\in F:D_f=k}\sum_{j\in P_{fD_f}}(w^{j,k}_{f,t}-w^{j,k}_{f,t-1})\leq A_k(t)\ \ \forall\ k\in K, t\in T \quad (4.3.3)$$

$$\sum_{f\in F:j\in S_f}\Big(\sum_{j'\in P_{fj}} w^{j'j}_{f,t}-\sum_{k\in N_{fj}} w^{jk}_{f,t}\Big)\leq S_j(t)\ \forall\ j\in S_f, t\in T \quad (4.3.4)$$

$$\sum_{j'\in P_{fj}} w^{j'j}_{f,t}\leq \sum_{j'\in P_{fj}}\sum_{i\in P_{fj'}} w^{ij'}_{f,t-l_{fj}}\ \forall\ f\in F, j\in S_f, t\in T^j_f, \quad (4.3.5)$$

$$\sum_{j'\in P_{fj}} w^{j',j}_{f,\overline{T}^f_j}-\sum_{k\in N_{fj}} w^{j,k}_{f,\overline{T}^f_k}=0\ \forall\ f\in F, j\in S_f: j\neq D_f \quad (4.3.6)$$

$$\sum_{k\in N_{fj}}\sum_{j'\in P_{fk}} w^{j'k}_{f,\overline{T}^f_k}\ \leq 1\ \forall\ f\in F, j\in S_f: j\neq D_f \quad (4.3.7)$$

$$w^{O_f,O_f}_{f,\overline{T}^f_j}-\sum_{j\in P_{fD_f}} w^{j,D_f}_{f,\overline{T}^f_j}=0\ \forall f\in F \quad (4.3.8)$$

$$w^{k,k}_{f,t}-\sum_{j\in P_{f'k}} w^{j,k}_{f',t-s'_f}\leq 0\ \forall\ (f',f)\in C, t\in T^k_f, k=O_f=D_{f'} \quad (4.3.9)$$

$$\sum_{j \in P_{fD_f}} w^{j,D_f}_{f,\overline{T}^f_{D_f}} \leq \sum_{j' \in P_{fD_{f'}}} w^{j',D_{f'}}_{f',\overline{T}^f_{D_{f'}}} \ \forall (f',f) \in C, \tag{4.3.10}$$

$$w^{O_f,O_f}_{f,t} - \sum_{j \in P_{fD_f}} w^{D_f}_{f,t+max_f} \leq 0 \ \forall \ f \in F, t \in T^f_{O_f} : t + max_f \in T^f_{D_f} \tag{4.3.11}$$

$$\sum_{j' \in P_{fj}} \left(w^{j'j}_{f,t} - w^{j'j}_{f,t-1} \right) \leq 0 \ \forall f \in F, j \in S_f, t \in T^f_j \tag{4.3.12}$$

$$w^{jj'}_{f,t} \in \{0,1\} \ \ \forall \ f \in F, j' \in S_f, j \in P_{fj'}, t \in T^j_f \tag{4.3.13}$$

Alternative Model Formulation: The complete formulation below is an alternative formulation based on the sector approach. The difference between this and the former model is that the formulation consists of both the first

$$\text{Minimise} \sum_{f \in F} \Big[\sum_{j \in P_{fj}} \sum_{t \in T^f_{D_f}} C^f_{td} \Big(w^{j,D_f}_{f,t} - w^{j,D_f}_{f,t-1} \Big) - \sum_{t \in T^f_{O_f}} C^f_{gr} \big(w^{O_f,O_f}_{f,t} - w^{O_f,O_f}_{f,t-1} \big) \Big] \tag{4.3.14}$$

Subject to

$$w^j_{f,t} = \sum_{j' \in N_{fj}} w^{j',j}_{f,t} \ \ \forall f \in F, j \in S_f, t \in T^f_j \tag{4.3.15}$$

$$\sum_{f \in F : O_f = k} (w^k_{f,t} - w^k_{f,t-1}) \leq D_k(t) \ \ \forall \ k \in K, t \in T \tag{4.3.16}$$

$$\sum_{f \in F : D_f = k} (w^k_{f,t} - w^k_{f,t-1}) \leq A_k(t) \ \ \forall \ k \in K, t \in T \tag{4.3.17}$$

$$\sum_{f \in F : j \in S_f} \Big(\sum_{j' \in P_{fj}} w^{j'j}_{f,t} - \sum_{k \in N_{fj}} w^{jk}_{f,t} \Big) \leq S_j(t) \ \forall \ j \in S_f, t \in T \tag{4.3.18}$$

$$\sum_{j' \in P_{fj}} w^{j'j}_{f,t} \leq \sum_{j' \in P_{fj}} \sum_{i \in P_{fj'}} w^{ij'}_{f,t-l_{fj}} \ \forall \ f \in F, j \in S_f, t \in T^j_f, \tag{4.3.19}$$

$$w^j_{f,\overline{T}^f_j} \leq \sum_{k \in N_{fj}} w^{jk}_{f,\overline{T}^f_k} \ \forall \ f \in F, j \in S_f : j \neq D_f, \tag{4.3.20}$$

$$\sum_{k \in N_{fj}} w^{jk}_{f,\overline{T}^f_k} \ \leq 1 \ \forall \ f \in F, j \in S_f : j \neq D_f \tag{4.3.21}$$

$$\sum_{j'\in P_{fj}} w^{j',j}_{f,\overline{T}^f_j} - \sum_{k\in N_{fj}} w^{j,k}_{f,\overline{T}^f_k} = 0 \ \forall\ f \in F, j \in S_f : j \neq D_f \tag{4.3.22}$$

$$w^{O_f}_{f,\overline{T}^f_j} - \sum_{j\in P_{fD_f}} w^{j,D_f}_{f,\overline{T}^f_j} = 0 \ \forall f \in F \tag{4.3.23}$$

$$w^{O_f}_{f,t} - w^{D_{f'}}_{f',t-s_f} \leq 0 \ \forall\ (f',f) \in C, t \in T^k_f : t - s_f \in T^{f'}_{O_{f'}} \tag{4.3.24}$$

$$\sum_{j\in P_{fD_f}} w^{j,D_f}_{f,\overline{T}^f_{D_f}} \leq \sum_{j'\in P_{fD_{f'}}} w^{j',D_{f'}}_{f',\overline{T}^f_{D_{f'}}} \ \ \forall (f',f) \in C, \tag{4.3.25}$$

$$w^{O_f}_{f,t} - w^{D_f}_{f,t+max_f} \leq 0 \ \forall\ f \in F, t \in T^f_{O_f} : t + max_f \in T^f_{D_f} \tag{4.3.26}$$

$$\sum_{j'\in P_{fj}} \left(w^{j'j}_{f,t} - w^{j'j}_{f,t-1}\right) \leq 0 \ \forall f \in F, j \in S_f, t \in T^f_j \tag{4.3.27}$$

$$w^j_{f,t}, w^{jj'}_{f,t} \in \{0,1\} \ \ \forall\ f \in F, j' \in S_f, j \in P_{fj'}, t \in T^j_f \tag{4.3.28}$$

Constraints (4.3.2), (4.3.3) and (4.3.4) as well as constraints (4.3.16), (4.3.17) and (4.3.18) are the capacity constraints that takes into consideration *the capacities of the different components of the system*. Constraints (4.3.15) in the alternative formulation describes *the relationship between the basic decision variables and the modified decision variables* (Bertsimas and Patterson, 1998).

Constraints 4.3.2 and 4.3.3 (4.3.16 and 4.3.17) ensure *that the number of flights departing from or arriving at an airport, k, at time period t does not exceed the departure and arrival capacity of airport k for that time period*. Constraints 4.3.4 (4.3.18) ensure that *the number of flights in a sector j at time period t does not exceed the sector capacity for that period*. Since for each sector k, we already know the previous sector j, the second term of Constraint 4.3.4 (4.3.18), $\sum_{k\in N_{fj}} w^{jk}_{f,t}$, takes 1 if and only if the first term, $\sum_{j'\in P_{fj}} w^{j'j}_{f,t}$ takes 1 and for this reason, the difference cannot take the value -1. Moreover, the first expression takes 1 if flight f has arrived at sector j at time t and if the flight crossed to one of the subsequent sectors, then the second term takes 1. This implies that flight f will not be counted among those flights that are still in sector j. Otherwise, $\sum_{k\in N_{fj}} w^{jk}_{f,t} = 0$ and the difference between the terms takes the value of 1 implying that flight f will be one of the flights that are still in sector j (Bertsimas et al., 2011).

Constraints 4.3.5, 4.3.6 and 4.3.7 (4.3.19, 4.3.20, 4.3.21 and 4.3.22) represents *the connectivity between sectors*. Constraint 4.3.5 (4.3.19) stipulates that *a flight is not permitted to enter the next sector on its route unless it has spent at least the minimum required travelling time through one of the previous sector on in its path*. This means a flight cannot arrive at sector j by time$t$ if it has not arrived at one of the previous sectors by time

$t - l_{fj}$(Bertsimas and Patterson, 1998; Bertsimas et al., 2011).

Constraint 4.3.6 (4.3.22) ensures that *the number of flights arriving in sector j must equal the number of flights leaving sector j*. Constraint 4.3.7 (4.3.20 and 4.3.21) stipulates that a *flight must arrive at one of the subsequent sectors by the latest time period at which it is allowed to reach these sectors* (Bertsimas et al., 2008, 2011).

Constraint 4.3.8 (4.3.23) ensures that *a flight gets to its destination if it departs from its origin airport and vice versa; otherwise, the flight is cancelled.* Constraint 4.3.9 (4.3.24) represents *flight connectivity and handle instances where an aircraft for a flight is scheduled to perform a subsequent flight within certain user specified time interval continued that is flight continuation where f', f, s_f denotes the first flight, subsequent flight and turnaround time*[2] Constraint 4.3.10 (4.3.25) represents *the flight cancellation priority such that the variable upper bound ensures that flight f is not cancelled if flight f' is not.* Constraint 4.3.11 (4.3.26) guarantees that *the flight arrives to its destination without exceeding the maximum acceptable duration of flight; otherwise the flight is cancelled.* Lastly, Constraint 4.3.12 (4.3.27) guarantees *connectivity in time for all flights.* Thus, if *a flight arrived at sector j by time $\tilde{t}$ to its destination, then the variables have to take a value 1 for all later time periods* $(t \geq \tilde{t})$ (Bertsimas et al., 2008, 2011).

4.3.2 IP model for the ATFMRP (Path-Based Approach)

The complete IP formulation based on the path approach, a variant of the basic ATFMP model, for reducing air traffic delay and congestion in en route airspace is presented below. The constraint interpretation for the sector approach still applies here except for slight changes. The constraints interpretation for both the main and alternative formulation were given simultaneously.

$$\text{Minimise} \sum_{f\in F}\Big[\sum_{t\in T^f_{D_f}} C^f_{td}\Big(w^{D_f,r}_{f,t} - w^{D_f,r}_{f,t-1}\Big) - \sum_{t\in T^f_{O_f}} C^f_{gr}\big(w^{O_f,r}_{f,t} - w^{O_f,r}_{f,t-1}\big)\Big] \tag{4.3.29}$$

Subject to

$$\sum_{f\in F:O_f=k} (w^{k,r}_{f,t} - w^{k,r}_{f,t-1}) \leq D_k(t) \;\; \forall\, k \in K, t \in T \tag{4.3.30}$$

$$\sum_{f\in F:D_f=k} (w^{k,r}_{f,t} - w^{k,r}_{f,t-1}) \leq A_k(t) \;\; \forall\, k \in K, t \in T \tag{4.3.31}$$

$$\sum_{f\in F:r(f,i)=j,r(f,i+1)=j',i\leq N_f} \Big(w^{j,r}_{f,t} - w^{j',r}_{f,t}\Big) \leq S_j(t) \; \forall\, j \in S, t \in T \tag{4.3.32}$$

[2] minimum amount of time required to prepare flight f for departure following the arrival of flight f'.

$$w^{O_f,r}_{f,\overline{T}^f_{O_f}} - w^{D_f,r}_{f,\overline{T}^f_{D_f}} = 0 \ \forall f \in F \tag{4.3.33}$$

$$w^{k,r}_{f,t} - w^{k,r'}_{f',t-s'_f} \leq 0 \ \forall \ (f',f) \in C, t \in T^k_f, k = r(f',N_{f'}) = r(f,1) \tag{4.3.34}$$

$$w^{D_f,r}_{f,\overline{T}^f_{D_f}} \leq w^{D_{f'},r'}_{f',\overline{T}^f_{D_{f'}}} \ \ \forall (f',f) \in C, \tag{4.3.35}$$

$$w^{O_f,r}_{f,t} - w^{D_f,r}_{f,t+max_f} \leq 0 \ \forall \ f \in F, t \in T^f_{O_f} : t + max_f \in T^f_{D_f} \tag{4.3.36}$$

$$w^{j,r}_{f,t} - w^{j',r}_{f,t+l(f,j)} \geq 0 \ \forall f \in F, r \in R_f, j = r(f,i), j' = r(f,i+1), i < N_f, t \in T^f_j \tag{4.3.37}$$

$$\sum_{r \in R_f, k = r(f,1)} w^{k,r}_{f,\overline{T}^f_k} = 1 \ \forall f \in F \tag{4.3.38}$$

$$w^{j,r}_{f,t} + w^{j',r'}_{f,t'} \geq 0 \ \forall f \in F, r, r' \in R_f, j \in r_f, j' \in r'_f, t \in T^f_j \tag{4.3.39}$$

$$w^{j,r}_{f,t} - w^{j,r}_{f,t-1} \geq 0 \ \forall f \in F, r \in R_f, j \in r_{nf}, t \in T^f_j \tag{4.3.40}$$

$$w^{j,r}_{f,t} \in \{0,1\} \ \ \forall \ f \in F, j \in r_{nf}, r \in R_f, t \in T^j_f \tag{4.3.41}$$

Alternative Model Formulation: The complete formulation below is an alternative formulation based on the path approach. The difference between this and the former model is that the formulation consists of both the first

$$\text{Minimise} \sum_{f \in F} \Big[\sum_{t \in T^f_{D_f}} C^f_{td} \Big(w^{D_f,r}_{f,t} - w^{D_f,r}_{f,t-1} \Big) - \sum_{t \in T^f_{O_f}} C^f_{gr} \big(w^{O_f,r}_{f,t} - w^{O_f,r}_{f,t-1} \big) \Big] \tag{4.3.42}$$

Subject to

$$w^j_{f,t} = \sum_{r \in R_f} w^{jr}_{f,t} \ \ \forall f \in F, j \in r_{nf}, t \in T^f_j \tag{4.3.43}$$

$$\sum_{f \in F : O_f = k} (w^k_{f,t} - w^k_{f,t-1}) \leq D_k(t) \ \ \forall \ k \in K, t \in T \tag{4.3.44}$$

$$\sum_{f \in F : D_f = k} (w^k_{f,t} - w^k_{f,t-1}) \leq A_k(t) \ \ \forall \ k \in K, t \in T \tag{4.3.45}$$

$$\sum_{f \in F: r(f,i)=j, r(f,i+1)=j', i \leq N_f} \left(w_{f,t}^{j} - w_{f,t}^{j'} \right) \leq S_j(t) \ \forall \ j \in S, t \in T \tag{4.3.46}$$

$$w_{f,\overline{T}_{O_f}^f}^{O_f} - w_{f,\overline{T}_{D_f}^f}^{D_f} = 0 \ \forall f \in F \tag{4.3.47}$$

$$w_{f,t}^{k,r} - w_{f',t-s'_f}^{k,r'} \leq 0 \ \forall \ (f', f) \in C, t \in T_f^k, k = r(f', N_{f'}) = r(f,1) \tag{4.3.48}$$

$$w_{f,\overline{T}_{D_f}^f}^{D_{f,r}} \leq w_{f',\overline{T}_{D_{f'}}^f}^{D_{f'},r'} \ \ \forall (f', f) \in C, \tag{4.3.49}$$

$$w_{f,t}^{O_f,r} - w_{f,t+max_f}^{D_f,r} \leq 0 \ \forall \ f \in F, t \in T_{O_f}^f : t + max_f \in T_{D_f}^f \tag{4.3.50}$$

$$w_{f,t}^{j,r} - w_{f,t+l(f,j)}^{j',r} \geq 0 \ \forall f \in F, r \in R_f, j = r(f,i), j' = r(f,i+1), i < N_f, t \in T_j^f \tag{4.3.51}$$

$$\sum_{r \in R_f, k=r(f,1)} w_{f,\overline{T}_k^f}^{k,r} = 1 \ \forall f \in F \tag{4.3.52}$$

$$w_{f,t}^{j,r} + w_{f,t'}^{j',r'} \geq 0 \ \forall f \in F, r, r' \in R_f, j \in r_f, j' \in r'_f, t \in T_j^f \tag{4.3.53}$$

$$w_{f,t}^{j,r} - w_{f,t-1}^{j,r} \geq 0 \ \forall f \in F, r \in R_f, j \in r_{nf}, t \in T_j^f \tag{4.3.54}$$

$$w_{f,t}^{j,r} \in \{0,1\} \ \ \forall \ f \in F, j \in r_{nf}, r \in R_f, t \in T_f^j \tag{4.3.55}$$

Constraints 4.3.30, 4.3.31 and 4.3.32 (4.3.44, 4.3.45 and 4.3.46) are *the capacity constraints that takes into consideration the capacities of the different components of the system.* As with the sector-based approach, constraints (4.3.43) in the alternative formulation describes the relationship between the basic decision variables and the modified decision variables.

Constraints 4.3.30 and 4.3.31 (4.3.44 and 4.3.45) ensure *that the number of flights departing from or arriving at an airport, k, at time period t does not exceed the departure and arrival capacity of airport k for that time period.* The difference between the terms in 4.3.30 and 4.3.31 (4.3.44 and 4.3.45) *equals one when the first term and second term is one and zero respectively. Thus, the differences capture the time at which a flight uses a given airport.* Constraint 4.3.32 (4.3.46) ensures that *the number of flights in a sector j at time period t does not exceed the sector capacity for that period.* This can be seen from the fact that the first term will have the value of 1 *if flight f has arrived in sector j by time t and second* *term will be 1 if flight f has arrived the next sector by time t.* Moreover, the only time the sum will have a value of 1 is if the flights have arrived at j and have not yet departed by time t. Thus, similar to the case of 4.3.30 and 4.3.31 (4.3.44 and 4.3.45), the difference captures the flight which are currently in sector j at time t (Bertsimas and Patterson, 1998; Bertsimas et al., 2011).

Constraint 4.3.33 (4.3.47) ensures that *a flight arrives its destination if it departs from its origin airport and vice versa; otherwise, the flight is cancelled.* Constraint 4.3.34 (4.3.48) ensures *connectivity between flights and handle instances where the flight's aircraft is scheduled to perform a subsequent flight within certain user specified time interval that is flight continuation* where $f', f, s_{f'}$ denotes the first flight, subsequent flight and turnaround time. Constraint 4.3.35 (4.3.49) represents *the flight cancellation priority such that the variable upper bound ensures that flight f is not cancelled if flight f' is not.* Constraint 4.3.36 (4.3.50) guarantees that *the flight arrives to its destination without exceeding the maximum acceptable duration of flight; otherwise the flight is cancelled* (Bertsimas et al., 2008, 2011).

Constraint 4.3.37 (4.3.51) guarantees *connectivity between sectors in a flight route. The interpretation is if flight f arrives at j' by time $t+l_{fj}$, then it must have arrived at j by time t where j' and j are adjacent sectors in the flight f's path.* More precisely, a flight is not allowed to enter the next sector on its path *if it has not spent the minimum required time for traveling through sector j, the current path, in its path.* Constraint 4.3.39 (4.3.53) guarantees that *exactly one route is chosen per flight while Constraint 4.3.38 (4.3.52) ensures that at most one route is chosen for every flight.* Constraints 4.3.38 (4.3.52) are important for *relaxations and are redundant for the IP formulation* as they are implied by Constraints 4.3.39 (4.3.53). Lastly, Constraint 4.3.40 (4.3.54) guarantees *connectivity in time for all routes.* Thus, *if a flight arrived at sector j by time $\tilde{t}$ along its route to its destination, then the variables has to take a value 1 for all later time periods* ($t \geq \tilde{t}$) (Bertsimas and Patterson, 1998; Bertsimas et al., 2008, 2011)

4.3.3 Size of the Formulation

The size of the formulation for the proposed models in the previous section can be determined given any data instance based on the generalised formula presented in Bertsimas and Patterson (1998) and defined below:

1. The *actual number of variables* is given by

$$\sum_{f} \sum_{j \in P_f} |T_f^j|, \tag{4.3.56}$$

since *each flight has a different number of sectors and number of feasible time inter-*

vals associated with it (Bertsimas and Patterson, 1998).

2. The *upper bound on the number of variables* w_{ft}^{j} can be calculated using the expression below

$$|F|DX, \text{ where } D = \max_{f \in F, j \in P_f} |T_f^j| \ \& \ X = \max_{f \in F} N_f \tag{4.3.57}$$

3. The *exact number of constraints* is

$$2|K||T| + |S||T| + 2\sum_{f \in F}\sum_{j \in P_f} |T_f^j| + \sum_{(f',f) \in C} \min\{|T_f^j|, |T_{f'}^k|\} \tag{4.3.58}$$

where $j = P(f,1)$ and $k = P(f', N_{f'})$.

4. The *upper bound on the number of constraints* can be calculated as

$$2|K||T| + |S||T| + 2|F|DX + |C|D. \tag{4.3.59}$$

Describing the notations in words,

- D is the *maximum cardinality of the set of feasible times for flight f to be in sector j taken over all f and j* (Bertsimas and Patterson, 1998).
- X is the *maximum number of sectors that a flight passes through its route, taken over all flights.* $X \geq 2$ since the origin and destination airports are counted as sectors on flights path (Bertsimas and Patterson, 1998).
- $|F|, |T|, |S|, |K|$ and $|C|$ represents the *total number of flights, time periods, sectors, airports and continued flights* respectively (Bertsimas and Patterson, 1998).

4.4 Chapter Summary

An insight into the proposed integer programming formulation for reducing air traffic delay and congestion was given in this chapter. The proposed formulations were described based on the sector and path based approaches proposed by Bertsimas and Patterson (1998) for including re-routing option in the BATFMP model. Besides extending the basic model to account for re-routing options, additional constraints like flight cancellation priorities and penalisation constraints were also incorporated into the formulation. Moreover, new cost functions to be minimised were also defined. The proposed formulation is as strong as the basic formulation because of the facet-defining constraints that were used. The additional index in the decision variables that were used in the formulation is the distinguishing factor between the proposed models and other models in literature that consider re-routing as a control option.

Chapter 5

Implementation and Computational Results

This chapter is dedicated towards the discussion and presentation of the numerical implementation of the basic ATFMP models on constructed input data sets and parameters considering different possible instances with fictitious airports and sectors. The intention is to solve the basic and modified models using the artificially constructed data sets and analyse the resulting solution. Most importantly, to ascertain the computational performance of the model based on the constructed data sets and verify if the constructed data sets support the claims and conclusions drawn in reviewed literature. The method of solution, as well as, the software specification, are briefly outlined in Section 5.1 while Section 5.2 provides a description of the computer specification, as well as, the artificially constructed datasets used for the implementation. Results and findings were presented in Sections 5.3 and 5.4 respectively.

5.1 Method of Solution

In this section, the method of solution used for implementation are outlined. The different stages of the implementation process and the software used were also described in details.

The optimisation model was solved using the *GLPK (GNU Linear Programming Kit) solver (version 4.57)* which comes in the form of a *callable library* and is solely designed for solving LP, MIP, and other related problems (Makhorin, 2015). The *GNU Mathprog Language (GMPL)*, a subset of the *AMPL modeling language*, was used to describe the models. This is because it uses *symbolic algebraic notation* which has advantages over other modelling languages. GNU MathProg language consists *a set of statements and data blocks* that must be in the required format and the implementation is based on the description presented in Fourer et al. (1987). More precisely, GNU Mathprog is *the model translator* that analyses the model description and translates it into internal data structures that can be used either for generating mathematical programming problem instance or directly by a solver to obtain numeric solution of the problem (Fourer et al., 1987; Makhorin, 2015).

For the implementation, the stand-alone solver, glpsol, within GLPK and CPLEX optimisation solver was used since CPLEX standalone solver is part of the GLPK together with the fact that CPLEX can be accessed through the GMPL. The solvers address the optimisation problem using the generated output by the GNU Mathprog translator or language processor.

There are different stages involved in the implementation of the model. First, the mathematical model is described using GMPL. The input format of the GMPL model consists of the following items sets, parameters, variables, objective, constraints, of which the sample format of the GMPL Models as well as the sample of the associated datasets for the different model used for this work can be located in the appendix section for reference purposes. See Fourer et al. (1987); Makhorin (2015) for more details on the model description.

The input statements for the different components of the model as described in Makhorin (2015) are highlighted below:

1. ***set*** *setname alias domain, attrib,* $\cdots$ *, attrib;*
2. ***param*** *parametername alias domain, attrib,* $\cdots$ *, attrib;*
3. ***var*** *variablename alias domain, attrib,* $\cdots$ *, attrib;* where
   - *setname* is a symbolic name of the set, parameter or variable;
   - *alias* is an optional string literal, which specifies an alias of the set, parameter or variable;
   - *domain* is an optional indexing expression, which specifies a subscript domain of the set, parameter or variable;
   - *attrib,* $\cdots$*, attrib* are optional attributes of the set, parameter or variable (Fourer et al., 1987; Makhorin, 2015).
4. The objective statement is
   a) ***minimise*** *name alias domain : expression;*
   b) ***maximise*** *name alias domain : expression;* where
   - *name* is a symbolic name of the objective;
   - *alias* is an optional string literal, which specifies an alias of the objective;
   - *domain* is an optional indexing expression, which specifies a subscript domain of the objective;
   - *expression* is a linear expression used to compute the linear form of the objective (Fourer et al., 1987; Makhorin, 2015)
5. The constraint statement is of the following form where the keyword ***subj to, s.t.*** can be used in place of ***subject to*** when writing the statement or omitted at all.
   i) ***subject to*** *name alias domain : expression , = expression;*

ii) ***subject to*** *name alias domain : expression , ≤ expression;*

iii) ***subject to*** *name alias domain : expression , ≥ expression;*

iv) ***subject to*** *name alias domain : expression , ≤ expression , ≤ expression;*

v) ***subject to*** *name alias domain : expression , ≥ expression , ≥ expression;* (Fourer et al., 1987; Makhorin, 2015).

The above information with the associated data sets will be used by the GMPL executable file when solving the optimisation problem. The stand alone solver, glpsol, is either *launched from the command line or from the shell* to solve models written in the GMPL using different command line options (Fourer et al., 1987; Makhorin, 2015). Solving the problem using CPLEX solver is a little bit different as the model description has to be in the required input format. In order for the IBM CPLEX optimisation solver to access the model, GNU Mathprog model for the problem has to be converted to LP model expressed in LP format, a required format for CPLEX, using the GNU/GLPK executable file. The lp file is thus read into the CPLEX solver in order to solve the model. As the model is being solved, the solution stages are displayed until an optimal solution is obtained. Once the final solution has been produced, the solutions pools are saved and readable solutions are generated from the pool (Fourer et al., 1987; Makhorin, 2015). See illustration of the result details in Section 5.3 and Appendix A for more clarification.

5.2 The Data

The computational performance of the basic models was tested on artificially constructed data sets for verification purposes. Three nominal data sets (Datasets 1 - 3) were used in validating the computational performance of the models considering, at least, two problem instances for each data set. Table 5.1 shows the computer specification that was used for the implementation.

Table 5.1: Computer specification

Computer brand	HP
Model	HP Pavilion g6
Processor	Intel(R) Core(TM) i3-3110M CPU
Processor	speed 1
Number of processors	2.40 GHz
Number of cores	4
RAM size	4 GB
Operating system	Windows 8 Enterprise
System type	64 bit Operating system

Data sets 1 - 3 were constructed with an imaginary airports and flights. The *total number of flights in each data set varied from 4 to 15* while the *time horizon varied from two* *hours to just over five hours discretised into time units of 15-minute or 30-minutes each.* The total number of airports considered ranges from 4 to 7 while the arrival, departure and enroute sector capacities vary from 1 to 5 flights per time period depending on the number of conflicting flights in the set which is assumed to be the same for all time periods. The percentage of connecting flights ranged from 15% to 75% depending on the instances considered. The feasible time periods, calculated during the preprocessing phase and defined for each of the instances, agrees with the scheduled departure and arrival timings. The ground and airborne delay cost was assigned to each flight in such a way that the airborne delay are higher than the ground delay cost.

Each of the instances consist of *a set of flights, airports, sectors, time periods, feasible time periods, scheduled arrival and departure times, system resources and enroute capacities, set of continued flights* and other relevant information. These data sets were constructed in such a way that there are flights having the same scheduled arrival, departure or sector crossing timing. Thereby giving rise to conflicting flights.

For each of the instances considered above, the primary data sets, that is, “the feasible time periods, the scheduled departure and arrival timings and other parameters” were modified to account for different scenarios considered in order to ascertain how the integrality of the solutions obtained are affected by the various parameters. The instances considered include: “when conflicts existed between flights, when flights departed or arrived earlier than the scheduled time, when outgoing flights departed before the arrival of incoming connecting flights, when flights were expected to traverse a particular sector at the same time period.

Each problem instance was solved using the stand alone solvers, glpsol within GLPK and the IBM ILOG CPLEX interactive optimiser 12.6.0 implemented on a PC with the specification described in Table 5.1 both for the LP relaxation and the integer programming problem.

As an illustration, given the departure and arrival airport for each flight, it is possible to estimate the demands on the airport for each time period based on the schedule departure and arrival timings. Figures 5.1, 5.2 and 5.3 are an illustration of such demands. Moreover, the demand for different sectors can be computed as well given the predefined or preferred route of each flight and the time for the flight to reach each sector. Fig 5.4 is an illustration of sector demands for randomly generated flights dataset.

The summary of the results for selected instances are tabulated in the next section with a brief explanation.

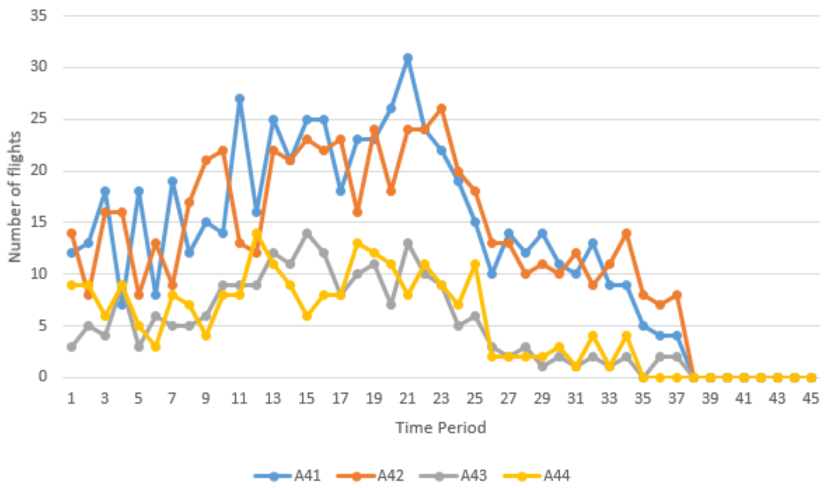


Figure 5.1: Total demand for airport usage at each time period.

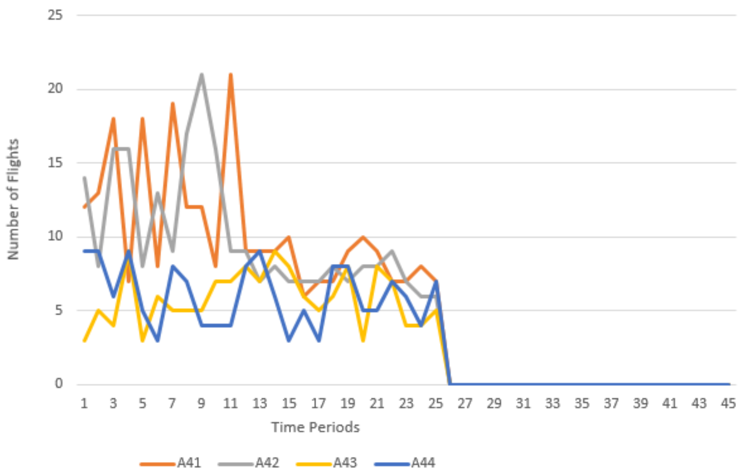


Figure 5.2: Airport demands for departure for each time period

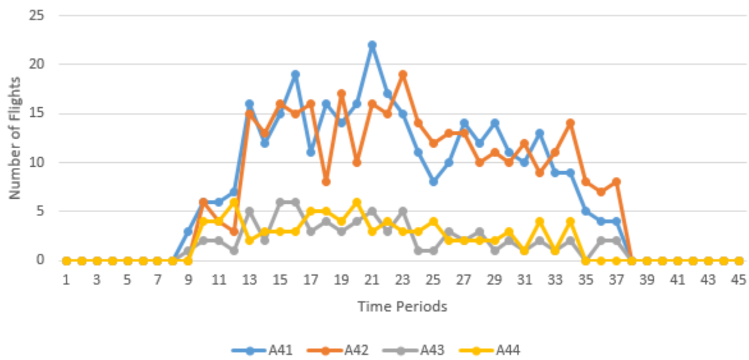


Figure 5.3: Airport demands for arrival for each time period

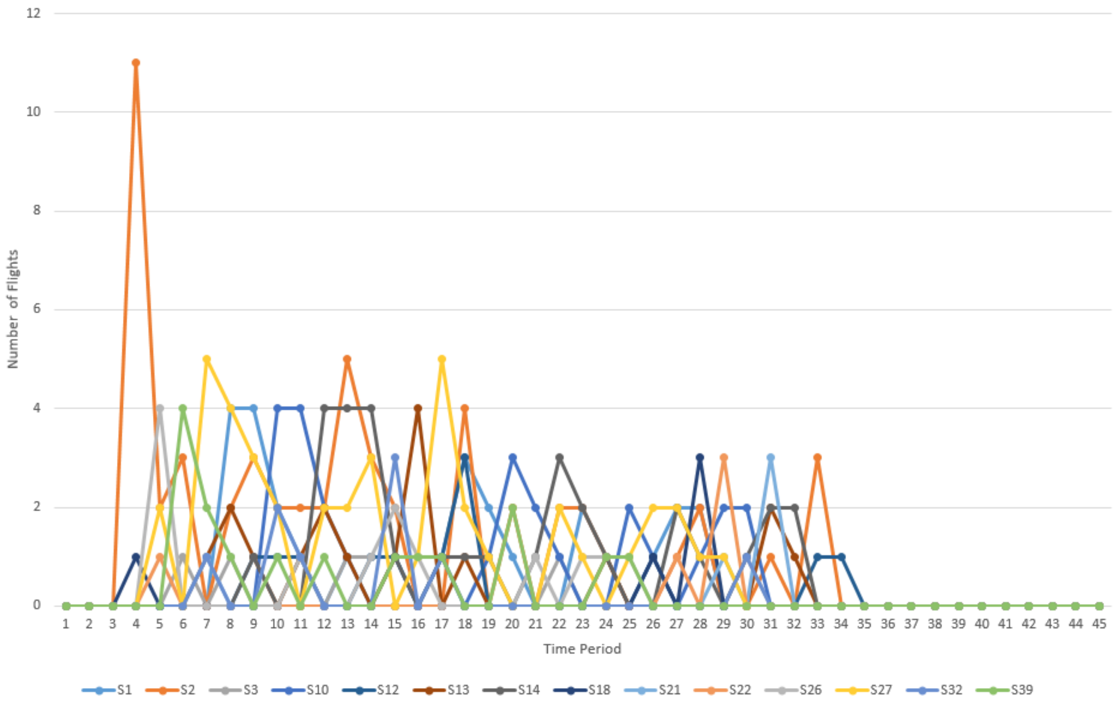


Figure 5.4: Sector demands at different time period.

5.3 Computational Results

Table 5.2 to 5.9 shows the summarised result of how delays were assigned to conflicting flights based on the different scenario considered for Data set 1 - 3. The experiments were tested using both the basic and modified ATFMP model with little or no changes in the input parameters and feasible time window.

Data set 1 was constructed based on the illustrative sample in Figure 3.1 presented in Chapter 3. The primary set consist of 4 airports, 6 sectors, 7 flights, time horizon of 4 hours discretised into 16 time periods of 15 minutes each. The sector transit time for each flights varies. Both ground and air delay costs are fixed for all flights.

From now onward, the headings of the table represents the following except otherwise stated. Gcost− ground delay cost, Acost− air delay cost, SDep− scheduled departure time, ADep− actual departure time, SArr− scheduled arrival time, AAr− actual arrival time, GD− ground delay, AD− air borne delay, TD− total delay, RGD− ground holding based on cost reduction. Flights are represented by their sectors or flight number denoted as $F6$.

For the result presented in Table 1, all flights are independent of each other. Flight 6 and 4 are conflicting with each other based on the scheduled arrival time. The result shows that flight 6 is delayed on the ground for 1 unit time period while flight 4 arrived at the destination airport without any delay since it has earliest departure time. This adheres to the "first in first out policy" in assigning delays to flight. The result in Table

Table 5.2: Results for dataset 1 without connecting flight

Flight	Gcost	Acost	SDep	ADep	SArr	AAr	GD	AD
1. *A-B-C-D*	R400	R20000	1	1	7	7	0	0
2. *A-F-E*	R400	R20000	1	1	6	6	0	0
3. *F-C-D*	R400	R20000	3	3	8	8	0	0
4. *A-B-C-E*†	R400	R20000	3	3	9	9	0	0
5. *D-E-C-F*	R400	R20000	3	3	10	10	0	0
6. *D-C-E*†	R400	R20000	5	6	9	10	1	0
7. *D-E-F-A*	R400	R20000	3	3	11	11	0	0

Arr Cap= 1; Dep Cap = 3 & Sec Cap = 3; Obj = 400 & Cost = 400, CPU = 5.3s

5.3 is still for dataset 1 with $F6$ continuing $F2$. The scheduled departure and arrival time remains the same as with previous one except for $F6$ which is changed in order to continue $F2$. The result showed how the flight was delayed for 2 time periods on the ground since its scheduled departure time was earlier than the arrival of the incoming flight. The implication of having the connecting flight depart before the arrival of the incoming flight is that passengers from the incoming flight are bound to miss their flights. To avoid such occurrences, the model ensures that the flights that need to be delayed in order to minimise costs are delayed. Delay will only be assigned to $F7$ if $F6$ was initially scheduled to depart at 7. Reducing the capacity of the system resources to allow only one (1) flight will result in delaying flights $F1, F3$ and $F7$ with a rapid increase in the number of iterations before obtaining optimal solution.

Table 5.3: Results for dataset 1 with connecting flight

Flight/Sectors	Gcost	Acost	SDep	ADep	SArr	AAr	GD	AD
1. *A-B-C-D*	R400	R20000	1	1	7	7	0	0
2. *A-F-E*	R400	R20000	1	1	6	6	0	0
3. *F-C-D*	R400	R20000	3	3	8	8	0	0
4. *A-B-C-E*	R400	R20000	3	3	9	9	0	0
5. *D-E-C-F*	R400	R20000	3	3	10	10	0	0
6. *E-C-B-A*	R400	R20000	5	7	9	11	2	0
7. *D-E-F-A*	R400	R20000	3	4	11	12	1	0

Arr Cap= 1; Dep Cap = 3 & Sec Cap = 3; Obj = 1200 & Cost = 1200, CPU = 6.9s

Data set 2 consists of 4 flights (2 independent flights) and 7 sectors. All the sectors are assumed to be containing airports. The horizon is 5 hours discretised into 11 time slots of 30 minutes each. The arrival capacity for each of the system resources is expected to accommodate at most 3 flights per time period.

For the solution presented in Table 5.5, none of the flights was delayed because there is

Table 5.4: Results for dataset 1 with connecting flight and capacity reduction

Flight/Sectors	Gcost	Acost	SDep	ADep	SArr	AAr	GD	AD
1. *A-B-C-D*	R400	R20000	1	2	7	8	1	0
2. *A-F-E*	R400	R20000	1	1	6*	6	0	0
3. *F-C-D*	R400	R20000	3	4	8	9	1	0
4. *A-B-C-E*	R400	R20000	3	3	9	9	0	0
5. *D-E-C-F*	R400	R20000	3	3	10	10	0	0
6. *E-C-B-A*	R400	R20000	5*	7	9	11	2	0
7. *D-E-F-A*	R400	R20000	3	4	11	12	1	0

Arr Cap= 1; Dep Cap = 1 & Sec Cap = 2; Obj = 1200 & Cost = 1200, CPU = 5.9s

no capacity reduction in the system. $F2$ would have been delayed if there was no capacity reduction in the arrival capacity. See Appendix A the detailed results.

Table 5.5: Results for data set 2 without conflicts

Flight/Sectors	Gcost	Acost	SDep	ADep	SArr	AAr	GD	AD
1. *DUR-LDY-MGH*	R300	R3000	1	1	3	3	0	0
2. *LDY-NCS-ULD*†	R700	R6000	3	3	5	5	0	0
3. *MGH-NCS*	R500	R4000	4	4	6	6	0	0
4. *MGH-RCB-ULD*†	R900	R6000	3	3	5	5	0	0

Arr Cap= 3; Dep Cap = 3 & Sec Cap = 3; Obj = 0 & Cost = 0, CPU = 3.7s

The next scenario that was considered is that of flights that actually depart or arrive before the scheduled time. A close observation in the basic model formulation shows that there is no restriction in the formulation that will prevent such from occurring. Therefore, there is a possibility of the solver assigning negative values to flights since the objective is to minimise costs. However, this was avoided by setting an auxiliary cost variable as $\gamma \geq \max(0, \alpha)$ to cater for cases where there is negative value in the cost variable by providing the real value. α represents the function to be minimised, that is, the computed cost where γ is the real cost. This scenario was avoided in the modified model as well as the new formulation by setting a restriction that $C^f_{tO_f} = 0$ for $t \leq d_f$ and $C^f_{tD_f} = 0$ for $t \leq a_f$.

To implement this scenario, the feasible set of time window for $F4$ was adjusted backwards so that the flight can depart before the scheduled time and arrive before time at the destination airport. The solution is presented in Table 5.6. There is no capacity reduction in the system.

Data set 3 is exactly the same as the previous one only that there is modification in the costs and flights. Moreover, the results presented were used to show how the modified model assigns delays to flights in comparison with the basic model. For the result presented

Table 5.6: Results for data set 2 with arrivals before scheduled times

Flight/Sectors	Gcost	Acost	SDep	ADep	SArr	AAr	GD	AD
1. *DUR-LDY-MGH*	R300	R3000	1	1	3	3	0	0
2. *LDY-NCS-ULD*	R700	R6000	3	3	5	5	0	0
3. *MGH-NCS*	R500	R4000	4	4	6	6	0	0
4. *MGH-RCB-ULD*	R900	R6000	3	1†	5	3	-2	0

Arr Cap= 2; Dep Cap = 2 & Sec Cap = 2; Obj =-1800 & Cost = 0, CPU = 2.7s

in Table 5.7, the indications with $*, \star, \dagger, \ddagger$ show the conflicts between the flights. That is, flights are anticipated to arrive or depart at the same time or that flights are traversing a sector at the same time period. $F1$ is continued by $F2$. Table 5.8 shows results of a scenario where there has been a change in the set of feasible time period for the flights while Table 5.9 shows the result for the instances with different values of α, the equivalence factor.

Table 5.7: Result for dataset 3 with conflicting flights

Flight/Sectors	Gcost	Acost	SDep	ADep	SArr	AAr	GD	AD
1. DUR-LDY*	400	20000	1	1	2	2	0	0
2.*LDY-NCS-ULD‡	600	30000	3	5	5	7	2	0
3. †MGH-RCB	800	40000	3	3	4	4	0	0
4. *RCB-ORT-ULD‡	1000	50000	3	3	5	5	0	0
5. $^\dagger, *$MGH-ORT-ULD‡	1200	60000	3	4	5	6	1	0

Flight/Sectors	SDep	ADep	SArr	AAr	TD	RGD
1. DUR-LDY*	1	1	2	2	0	0
2.*LDY-NCS-ULD‡	3	4	5	6	1	1
3. †MGH-RCB	3	3	4	4	0	0
4. *RCB-ORT-ULD‡	3	3	5	5	0	0
5. $^\dagger, *$MGH-ORT-ULD‡	3	5	5	7	2.00139	1

Arr Cap = Dep Cap = Sec Cap = 1; Obj1 = 2200 = Obj2 & Cost =2200, CPU = 5.3s

Table 5.8: Result for dataset 3 with different sets of feasible time window

Flight/Sectors	Gcost	Acost	SDep	ADep	SArr	AAr	GD	AD
1. DUR-LDY*	400	20000	1	1	2	2	0	0
2.*LDY-NCS-ULD‡	600	30000	3	4	5	7	1	1
3. †MGH-RCB	800	40000	3	3	4	4	0	0
4. *RCB-ORT-ULD‡	1000	50000	3	3	5	5	0	0
5. $^\dagger, *$MGH-ORT-ULD‡	1200	60000	3	4	5	6	1	0

Arr Cap= 1; Dep Cap = 1 & Sec Cap = 1; Obj1 = 31600 & Cost =2200, CPU = 5.5s

Table 5.9: Result for dataset 3 with different equivalence factor value

Flight/Sectors	SDep	ADep	SArr	AAr	GD	AD
1. DUR-LDY*	1	1	2	2	0	0
2.*LDY-NCS-ULD‡	3	4	5	6	1	1
3. †MGH-RCB	3	3	4	4	0	0
4. *RCB-ORT-ULD‡	3	3	5	5	0	0
5. †, $*$MGH-ORT-ULD‡	3	5	5	7	2.00139	1
Flight/Sectors	SDep	ADep	SArr	AAr	TD	RGD
1. DUR-LDY*	1	1	2	2	0	0
2.*LDY-NCS-ULD‡	3	4	5	6	1	0
3. †MGH-RCB	3	3	4	4	0	0
4. *RCB-ORT-ULD‡	3	3	5	5	0	0
5. †, $*$MGH-ORT-ULD‡	3	5	5	7	2.00139	0

Arrival Capacity = 1 at all times, OBJ1-12.013, OBJ2-3.013, CPU = 4.6s

5.4 Result Findings

The computational performance reveals that the problem size of the formulation is very huge even for simple data instances. The IBM ILOG CPLEX 12.6.0. interactive solver can compute optimal solution in relatively fast runtime than the glpsol LP/MIP solver and in almost all the cases considered, integral solutions were obtained from the LP relaxations of the formulations. This shows that the computational results reported in Bertsimas and Patterson (1998); Bertsimas et al. (2008, 2011) which were highlighted in Chapter 3 were not based on the specific data sets that were used. Integral solutions as can be seen, can also be obtained from our constructed datasets which are neither affected by the changes made in the input parameter nor the problem size. For all the instances considered, the resulting solutions were obtained within an optimality value gap of 1%. This affirms the fact that the formulation is a strong one since a lower value gap indicates a stronger model and the reason for choosing it as the underlying model for the proposed formulation. Furthermore, it was observed that the number of iterations and the run time required for optimal solution to be obtained varies with the capacity reductions and flight connectivity.

5.5 Chapter Summary

The basic ATFM models were solved using fabricated data sets in order to ascertain the performance of the resulting solution with respect to the constructed data sets. The implementation details, the data sets used and a summary of the results were given. Different scenarios were considered for each dataset and solved using the stand alone optimisation solvers GLPSOL and CPLEX. The computational result shows that CPLEX can solve each of the problem very fast time. Results also indicate that in almost all the cases considered, the solution was completely integral.

Chapter 6

Summary, Conclusion and Future Work

6.1 Summary

Can the Air Traffic Flow Management Problem be addressed? If so, what approaches and methods can be used to address it? What has been done in previous studies, and how can this be improved to manage the problem? These and other question formed the basis of this thesis. The research investigated the ATFMP and critically examined the existing models and optimisation techniques that have been used to address the problem. The research was conducted from an empirical and practical point of view in order to identify the best available optimisation models and solution approaches, and to improve on them.

The thesis was made up of five chapters. In the first chapter, a detailed insight on the ATFMP was provided. The researcher commenced by outlining the specific objectives and challenges of ATFM, thereby distinguishing between ATC and ATFM, the two components of Air Traffic Management system. An overview of the current practices and ATFM techniques implemented by the South African Central Airspace Management Unit to address the ATFM challenges in South Africa was also given.

Chapter 2 provides the conceptual framework for this study wherein the concept of modeling and optimisation, optimisation process and techniques as well as key ideas in integer programming and polyhedral combinatorics were extensively discussed.

Chapter 3 focused on optimisation models and approaches for solving the ATFMP both for the deterministic and stochastic case. These were critically analysed with particular focus on airport capacity and airspace allocation models. The general assumption of the existing ground holding approaches is that congestion problems and capacity constraint occur mainly at the terminal hence the flow of air traffic is not affected by the en-route airspace capacities. This however is a misconception. Research indicates that there are some countries where en-route airspace pose great challenges in managing the flow of air traffic. Moreover, the existing models classified under the ATFM approaches especially the ATFMRP and ATFMRP under uncertainty tend to be more reliable and viable in providing solutions to the current ATFM challenges in comparison to the current techniques being employed by traffic managers.

Having reviewed the existing optimisation approaches for solving the ATFMP, In Chapter 4, an insight into the proposed integer programming formulation for reducing air traffic delay and congestion problems in en-route airspace was provided. The IP formulation was based on the sector and path approaches proposed by Bertsimas and Patterson (1998) for including re-routing option in the BATFMP model. In order to extend the model to account for several variation, key ideas and basic assumptions relevant for the mathematical formulation were established. The objective function was also modified, set of feasible time window for a flight to arrive in a particular sector and the set of possible routes that a flight will choose from explicitly defined. Additional constraints like flight cancellation priorities, flow conservation and penalisation constraints were also incorporated into the formulation in addition to the re-routing options. Cost functions to be minimised were also defined with respect to the new decision variables. The distinguishing factor between the proposed formulation and other ones in literature is that an additional index was added in the decision variables that were used in the formulation. This led to the linearisation of some of the non-linear constraints found in other models.

Finally, the method of solution, description of the software used for implementation, details of the data sets used, as well as, the problem instances considered, the computational results and findings were presented in Chapter 5. The basic ATFMP model was validated using artificially constructed datasets with different scenarios programmed in GNU Mathprog Language (GMPL) and implemented on a HP Pavilion g6 Intel Core i3-3110M PC and solved with GLPK/GLPSOL and IBM ILOG CPLEX 12.6.0. interactive solvers with the goal of investigating the claims and conclusions drawn from the literature review of related work.

6.2 Conclusion

It emanated from this study, precisely in Chapter 3 that the models in Bertsimas and Patterson (1998); Bertsimas et al. (2008, 2011), which form the underlying model for other existing models and the proposed formulation, can be combined with the current practices for effective air traffic flow management and decision making. This is because the polyhedral structures, valid inequalities and facet-defining constraints used in the formulation makes it stronger, leading to its outstanding performance when tested on real-life data instances. The only limitation for adapting most optimisation models in practice is that most issues are yet to be fully considered as part of ATFM activities when formulating the mathematical models that address the problem. These issues include equity and fairness consideration, interaction with airlines, collaboration decision making with stakeholders, dynamic updating of decisions, size of the formulation and other ATFM challenges peculiar to areas. There is therefore need to explore more on these issues so as to bridge the gap that exists between research and practice.

The study further revealed the need for researchers to shift their focus from American and European countries and extend it to other developed, developing and underdeveloped countries where similar if not worse problem are encountered. Such a move will immensely help in the decision making of the countries considered and contribute positively to their economic growth and development.

It emerged that the proposed formulation presented in Chapter 4 is foreseen to be strong as the underlying model because of the facet-defining constraints that were used. The computational performance of the basic ATFMP models shows that the resulting solutions were completely integral in all the cases considered. This affirms the fact that the results reported in literature are not based on the specific data sets used but based on the strong formulation of the model which also yields integral solution for the artificially constructed datasets.

The computational performance further revealed that CPLEX 12.6.0. interactive optimisation solver can compute optimal solution even for large data instances in relatively fast run time when compared to the stand alone glpsol LP/MIP solver due to its special capabilities and characteristics. Changes in the input parameters does not in anyway affect the integrality of the solution. Observation from the computational results indicates that the resulting solutions were obtained within an optimal value gap of 1% and the number of iterations as well as the run time required to obtain the integral solutions increases exponentially with the size of the problem , the degree of connectivity varies when there is capacity reduction in the system resources.

ATFMP, as has been in reviewed literature is an NP-hard problem given its complexity. However, from the review and results presented herein, it is evident that both small and large-scale instances of ATFMP can be solved to optimal solution within the shortest possible computational times using either exact or heuristic optimization techniques after careful examination of the underlying structures of the formulation.

In conclusion, the study revealed that the ideas and underlying principles employed in the ATFMP formulation is not restricted only to the air transportation industry but can also be applied in other areas like manufacturing and road transportation industries, where *goods and services are flowing through a dynamical system with different types of capacitated elements*. Furthermore, both the stochastic and the deterministic ATFM solutions play an important role in the efficient operation of the air traffic system since in a practical implementation, the stochastic ATFM problem will be used to determine a strategic operational plan in the presence of uncertain forecasts while the deterministic ATFM problem will provide the tactical plan as new information is obtained.

6.3 Directions for Future Work

Having been restricted from accessing the datasets that have been in most studies found in literature as well as the license to most of the optimisation solvers, the implementation of the proposed formulation for reducing delay and congestion problem, as well as, comparison of results with other researcher results was impossible. It has thus been shelved for future work so as to allow for more time in validating and testing the robustness of the model. Moreover, the integrality of the two formulations based on each of the approaches will be investigated via the polyhedral structures in order to ascertain the computational performance. This decision will allow for more time to incorporate different modeling variations like interdependence between arrival and departure runways, Hub Connectivity with Multiple Connections, Banking of Flights Constraints, Equity and Fairness, Collaborative Decision Making between shareholders, interaction with airlines and dynamic updating of the solution into the proposed formulation. Moreover, extending the investigation as a future endeavour will allow the researcher ample time to come up with a good solution approach for solving the problem either by considering Lagrangian relaxation techniques together with any heuristics or with any other techniques. The deterministic model will also be extended to probabilistic environment to accommodate uncertainty. Thus, formulating a stochastic version of the resulting formulation

Based on the computational results obtained from the artificially constructed data sets, the size of the problem can be reduced by carefully examining the scheduled departure and arrival times. One method of doing such is to solve the problem only for the conflicting flights. This will automatically give rise to definition of two new sets say $\overline{F}$ and $F^{'}$ where $F^{'}$ and $\overline{F}$ is the conflicting and non-conflicting sets respectively. The process will then be solved for the conflicting sets. That is, to check if there exist conflicts among the flights, update the set of conflicting sets, check again for conflicts between both sets, update solution again and continue with the process until there is no longer conflicting flights. Thus, the future can concentrate on coming up with a good heuristic that will be able to do such an iterative process. This however may have its limitations and shortfalls which cannot be ascertained because it has not been implemented. This will lead to exploration of heuristics, meta-heuristic and other hybrid heuristics techniques in order to solve the ATFMP and make comparison with other existing heuristic and exact techniques.

In view of the current trends in South Africa as well as the fact that Africa has a sparse network, emphasis should be placed on Air Traffic Flow Management Rerouting Problem under uncertainty and in cases where disruptions affects the traffic flow. With this approach, the problem can be reformulated as a minimum cost multi-commodity network flow model on a Time-Space Network with Network Simplex Method as the solution approach. To consider rerouting of flights when there is uncertainty, search methods and heuristic techniques can be considered. It is also possible to consider what happens when a flight is rerouted to an airport that is not in the defined set of airports when uncertainty happens.

Appendix A

Computational Results

The purpose of presenting this computational result is to illustrate how GLPK and CPLEX solves optimisation problem using the same model and data instances. The detailed result is just for one instance.

GLPK/CPLEX Results for Dataset 1 Table 5.2

```
GLPK Integer Optimizer, v4.57
1079 rows, 688 columns, 1334 non-zeros
672 integer variables, all of which are binary
Preprocessing...
61 hidden packing inequaliti(es) were detected
8 hidden covering inequaliti(es) were detected
148 rows, 102 columns, 378 non-zeros
86 integer variables, all of which are binary
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  2.000e+04  ratio =  2.000e+04
GM: min|aij| =  3.761e-01  max|aij| =  2.659e+00  ratio =  7.071e+00
EQ: min|aij| =  1.414e-01  max|aij| =  1.000e+00  ratio =  7.071e+00
2N: min|aij| =  9.766e-02  max|aij| =  1.221e+00  ratio =  1.250e+01
Constructing initial basis...
Size of triangular part is 148
Solving LP relaxation...
GLPK Simplex Optimizer, v4.57
148 rows, 102 columns, 378 non-zeros
      0: obj =  -2.800000000e+05 inf =   5.880e+02 (24)
    102: obj =   3.600000000e+03 inf =   2.517e-14 (0)
*   114: obj =   4.000000000e+02 inf =   1.373e-14 (0)
OPTIMAL LP SOLUTION FOUND
Integer optimization begins...
+   114: mip =     not found yet >=              -inf        (1; 0)
+   114: >>>>>   4.000000000e+02 >=   4.000000000e+02   0.0% (1; 0)
+   114: mip =   4.000000000e+02 >=     tree is empty   0.0% (0; 1)
INTEGER OPTIMAL SOLUTION FOUND
Time used:   5.3 secs
Memory used: 1.6 Mb (1673269 bytes)
```

```
GLPSOL Solution File (Summary)
Problem:    alexdataset1a
Rows:       1079
Columns:    688 (672 integer, 672 binary)
Non-zeros:  1334
Status:     INTEGER OPTIMAL
Objective:  domy = 400 (MINimum)

CPLEX> read dataset1a.lp
Problem 'dataset1a.lp' read.
Read time = 0.00 sec. (0.08 ticks)
CPLEX> optimize
Tried aggregator 4 times.
MIP Presolve eliminated 983 rows and 619 columns.
Aggregator did 33 substitutions.
Reduced MIP has 62 rows, 36 columns, and 152 nonzeros.
Reduced MIP has 36 binaries, 0 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.00 sec. (1.06 ticks)
Found incumbent of value 60400.000000 after 0.00 sec. (1.21 ticks)
Probing time = 0.00 sec. (0.06 ticks)
Tried aggregator 1 time.
Reduced MIP has 62 rows, 36 columns, and 152 nonzeros.
Reduced MIP has 36 binaries, 0 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.00 sec. (0.10 ticks)
Probing time = 0.00 sec. (0.06 ticks)
Clique table members: 93.
MIP emphasis: balance optimality and feasibility.
MIP search method: dynamic search.
Parallel mode: deterministic, using up to 4 threads.
Root relaxation solution time = 0.00 sec. (0.08 ticks)

        Nodes                                               Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                        60400.0000   -98000.0000       18  262.25%
*     0+    0                          400.0000   -98000.0000       18     ---
      0     0        cutoff            400.0000      400.0000       18    0.00%
      0     0        cutoff            400.0000      400.0000       18    0.00%
Elapsed time = 0.02 sec. (1.65 ticks, tree = 0.00 MB, solutions = 2)

Root node processing (before b&c):
```

```
  Real time             =    0.02 sec. (1.67 ticks)
Parallel b&c, 4 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.02 sec. (1.67 ticks)

Solution pool: 2 solutions saved.

MIP - Integer optimal solution:  Objective = 4.0000000000e+002
Solution time =    0.02 sec.  Iterations = 18  Nodes = 0
Deterministic time = 1.67 ticks  (104.56 ticks/sec)
```

Appendix B

GNU Mathprog (GMPL) sample code

The sample codes presented here are for illustration purposes on how the BATFMP models can be represented using GMPL. They are not the codes used for implementation.

B.1 Sample Code I

```
# SAMPLE GMPL CODE FOR THE BATFMP MODEL
### SETS ###
set NU; # nat. nombers
set AS; #airports \& sectors
set FL; #Flights
set TI; #Time periods

### PARAMETERS ###
param arrcap{a in AS, t in TI}; #capacity 1-ARR
param seccap{a in AS, t in TI}; #capacity 2-SEC
param depcap{a in AS, t in TI}; #capacity 3-DEP
param atsect{f in FL, a in AS, t in TI};
#last time period for flight to be in sector
param etd{f in FL}; #estimated time of departure
param eta{f in FL}; #estimated time of arrival
param trt{f in FL, a in AS}; #transit time
param wtime{f in FL}; #Flight turnaround time
param gcost{f in FL}; #cost of holding flight on  ground per unit
param acost{f in FL}; #cost of air borne holding
param twind{t in TI, a in AS, f in FL}; #set of feasible time window
param cflight{g in FL, f in FL}; #continuing flights
#param if is 1 and set if its many
param sect_od{f in FL, n in NU}symbolic; #ith sector in flight route
param cad{f in F}; #set of flights sector cardinality

### VARIABLES ###
var w{f in FL, t in TI, a in AS} integer >= 0, <=1; #decision variable
var gdel{f in FL}; # ground delay variable
var adel_fl{f in FL}; #air delay variable
```

```
var cost; #variable cost = max (0, real cost)
var domyobj; #Domy obj func

### OBJECTIVE FUNCTION AND CONSTRAINTS ###
minimize domm: domyobj; #obj func
subject to domyobj1: domyobj = sum{f in FL}
(gcost[f]*gdel[f] + acost[f]*adel[f]);
subject to ps: cost >=0;
subject to ps1: cost >=domyobj; # the obj functn
subject to gdel{f in FL}:gdel[f]=sum{t in TI:twind[t, sect_od[f,1],f] = 1}
t*(w[f,t,sect_od[f,1]] - (if t-1 in TI then w[f,t-1,sect_od[f,1]]))
- etd[f]; #ground delay
subject to adel{f in FL}: adel[f] =
sum{t in TI: twind[t, sect_od[f,cad[f]],f] = 1}t*(w[f,t,sect_od[f,cad[f]]]
- (if t-1 in TI then w[f,t-1,sect_od[f,cad[f]]]))
-eta[f] - gdel[f]; #air delay
subject to deparrv{f in FL,t in TI,a in AS:twind[t,a,f] = 0.0}:w[f,t,a]=0;
#no arval/dep at infeasible time window
s.t. AtsectA{f in FL, a in AS, t in TI: atsect[f,a,t] = 1}: w[f,t,a] = 1;
#flight must arrive at  sector by time t

### CONSTRAINTS###
subject to departcap{t in TI, a in AS}:
sum{f in FL: a=sect_od[f,1] and twind[t,a,f]=1}(w[f,t,a]
-(if t-1 in TI then w[f,t-1,a])) <= depcap[a,t]; #capacity of orig airport
subject to arrvcap{t in TI, a in AS}:
sum{f in FL: twind[t,a,f]=1 and a = sect_od[f,cad[f]]}
(w[f,t,a]-(if t-1 in TI then w[f,t-1,a])) <= arrcap[a,t]; #cap (arrival)
subject to sectorcap{t in TI, a in AS}:
sum{f in FL, n in NU:n<cad[f] and twind[t,a,f]=1 and sect_od[f,n]=a}
(w[f,t,sect_od[f,n]] - w[f,t,sect_od[f,n+1]])<=seccap[k,t]; #capa (sector)
s.t. secconet{f in FL, n in NU, t in TI:n<cad[f] and twind[t,sect_od[f,n]
,f]=1}:(if t + trt[f,sect_od[f,n]] in TI then w[f,t + trt[f,sect_od[f,n]],
sect_od[f,n+1]] - w[f,t,sect_od[f,n]])<= 0; #connectivity btwn sectors
s.t.turnarrtime{f in FL, g in FL, t in TI: twind[t,sect_od[f,1],f]=1 and
cflighht[g,f]=1}:(w[f,t,sect_od[f,1]]- w[g,t-wtime[g],sect_od[g,cad[g]]])
<=0; #flight turnaround/waiting time
s.t. arrive{f in FL, a in AS, t in TI: twind[t,a,f]=1}:(w[f,t,a] -
(if t-1 in TI then w[f,t-1,a])) >= 0; ###
```

B.2 Sample Code II

```
# SAMPLE GMPL CODE FOR THE ATFMP MODEL

### PARAMETERS ###
param T > 0 integer; #number of time periods
param F > 0 integer; #number of flights
param S >=0 integer; #number of sectors

### SETS ###

set Airports; # set of airports
set OD within {i in Airports, j in Airports: i <> j};
#set of origin-destination pairs
set Sect_od{i in Airports, j in Airports};
set next{i in Airports, j in Airports, s in Sect_od[i,j]};
# set of subsequent sectors for each sector picked
set previous{i in Airports, j in Airports, s in Sect_od[i,j]};
#set of previous sectors for each sector picked
set CFlights dimen 2;
set CFLights within {1..F,1..F}; #Continuing flights


### PARAMETERS CONTD #########
param arrcap{k in Airports, t in 1..T}; #capacity (arrival)
param depcap{k in Airports, t in 1..T}; #capacity (departure)
param Sec_Capacity{j in 0..S, t in 1..T}; #sector capacity
param etd{f in 1..F}; #estimated departure time
param eta{f in 1..F}; #estimated arrival time
param orig{f in 1..F}; # airport of departure
param dest{f in 1..F}; # airport of arrival
param ftime{j in 0..S, l in 0..S}; #flight time from one element to
another element #default value set to 2
param time2sect{k in Airports, s in 0..S}; #flight time from an airport
to a sector of the ATFM system
param epsylon; #coefficient
param afa; #equivalence factor
param G_f{f in 1..F}; #max ground holding time
param A_f{f in 1..F}; #max air holding time
param arrsect{f in 1..F, j in Sect_od[orig[f],dest[f]]}:=  etd[f] +
time2sect[orig[f],j]; #schedule arrival time period to sector
```

```
param tun{f in 1..F}; #s_f, turnaround time for an aircraft

################### Calculated sets#######################
set Td1{f in 1..F} := {t in 1..T: t >= etd[f]};
set Td2{f in 1..F} := {t in 1..T: t <= min(etd[f]+ G_f[f],T)};
set Td{f in 1..F} := {Td1[f] inter Td2[f]}; #set of time window for dep
set Ta1{f in 1..F} := {t in 1..T: t >= eta[f]};
set Ta2{f in 1..F} := {t in 1..T: t <= min(eta[f] + G_f[f] + A_f[f], T)};
set Ta{f in 1..F} := {Ta1[f] inter Ta2[f]} ;#set of time window for arr
set Tj1{f in 1..F, j in Sect_od[orig[f],dest[f]]} := {t in 1..T: t
>= arrsect[f,j]};
set Tj2{f in 1..F, j in Sect_od[orig[f],dest[f]]} := {t in 1..T: t <=
min(arrsect[f,j] + G_f[f] + A_f[f], T)};
set Tj{f in 1..F,j in Sect_od[orig[f],dest[f]]}:={Tj1[f,j] inter Tj2[f,j]};
#set of feasible time period to be in sectors

######################## VARIABLES #########################
### VARIABLES ###
var w{f in 1..F, t in 1..T, k in 0..S} integer >= 0, <=1; #decsn variables
var totdel_fl{f in 1..F}; ##
var reddel_fl{f in 1..F}; ##
var cost; ###
var domyobj; #dummy obj functn

############ OBJECTIVE FUNCTION AND CONSTRAINTS ###########

minimize dommy: domyobj; #objective function
s.t. pos1: cost >=domyobj; # the objective function

#################Objective Main#########################
s.t. totdel{f in 1..F} : totdel_fl[f] = sum{t in Ta[f]}(if t >= eta[f]
then afa*((t - eta[f])**(1+epsylon)))*(w[f,t,dest[f]] - (if t-1 in Ta[f]
then w[f,t-1,dest[f]]));
s.t. redel{f in 1..F} : reddel_fl[f] = sum{t in Td[f]}(if t >= etd[f]
then (afa-1)*((t - etd[f])**(1+epsylon)))*(w[f,t,orig[f]]-(if t-1 in Td[f]
then w[f,t-1,orig[f]]));
##############newest##########################
s.t. domyobj1: domyobj = sum{f in 1..F}(totdel_fl[f] - reddel_fl[f]);

############## CONSTRAINTS###########
subject to departcap{t in 1..T, k in Airports}: sum{f in 1..F: k=orig[f]}
```

```
(w[f,t,k]-(if t-1 in 1..T then w[f,t-1,k])) <= depcap[k,t];#cap of depairpt
subject to arrivecap{t in 1..T, k in Airports}: sum{f in 1..F: k = dest[f]}
(w[f,t,k]-(if t-1 in 1..T then w[f,t-1,k]))<= arrcap[k,t]; #arrival capcity
subject to seccapa{t in 1..T, j in 0..S}:sum{f in 1..F:j in Sect_od[orig[f],
dest[f]] and j <> dest[f] }((w[f,t,j]-(sum{i in next[orig[f],dest[f],j]}
w[f,t,i]))) <= Sec_Capacity[j,t]; #sector capacity
subject to secconet{f in 1..F, j in Sect_od[orig[f],dest[f]], t in Tj[f,j]:
j <> orig[f]}:(w[f,t,j]) <= sum{i in previous[orig[f],dest[f],j]}
(if t-ftime[i,j] in Tj[f,j] then w[f,t-ftime[i,j],i]);#cant arrive if hasnt
subject to secconet2{f in 1..F, j in Sect_od[orig[f],dest[f]]: j <> dest[f]}:
(w[f, min(arrsect[f,j] + G_f[f] + A_f[f], T),j]) <= sum{i in next[orig[f],
dest[f],j]}(w[f, min(arrsect[f,i] + G_f[f] + A_f[f], T),i]);#arrv at subsqnt
subject to secconet3{f in 1..F, j in Sect_od[orig[f],dest[f]]: j <> dest[f]}:
sum{i in next[orig[f],dest[f],j]}(w[f,min(arrsect[f,i]+G_f[f]+A_f[f],T),i])
<=1; #must arrive by the latest time period
subject to continud{(f,g) in CFlights, t in Td[f]}:(w[f,t,orig[f]] -
(if t -tun[f] in Td[g] then w[g,t-tun[f],dest[g]])) <= 0; #continued flight
subject to arrvby{f in 1..F, j in Sect_od[orig[f],dest[f]], t in Tj[f,j]}:
(if t-1 in Tj[f,j] then w[f,t-1,j]) - w[f,t,j]  <= 0; #time conectivity
############################
```

Appendix C

Sample Data Sets

The data presented here are sample data instances to illustrate the required format for data to be accessed by models written in GMPL. They are not the ones used for implementation.

C.1 Sample Dataset I

```
#Generated data instances for ATFMP
data;
set Airports := 41 42 43 44 45;
set Sectors := 41 42 43 44 45 46 47; #41 42 43 44 DUR ULD RCB LDY NCS MGH;
set Flights := 1 2 3 4;  #1 2 6; #3 #4 5; #7 8 9 10 11 12;
set Times := 0 1 2 3 4 5 6 7 8 9 10; #0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15;

set Sect_od[41,44] := 41  43 44;
set Sect_od[44,45] := 44  45;
set Sect_od[43,42] := 43  46 42;
set Sect_od[44,42] := 44 47 42;

param Twindow default 0.0 :=
 [*,*,1]:    41 42 43 44 45 46 47 := #Calculated Set of feasible time window
0  .  .  .  .  .  .  .
1   1  .  .  .  .  .  .
2   1  .  1  .  .  .  .
3   1  .  1  1  .  .  .
4   1  .  1  1  .  .  .
5   1  .  1  1  .  .  .
6   1  .  1  1  .  .  .
7   1  .  1  1  .  .  .
8   .  .  1  1  .  .  .
9   .  .  1  1  .  .  .
10  .  .  1  1  .  .  .
[*,*,3]:    41 42 43 44 45 46 47 :=
0  .  .  .  .  .  .  .
1   .  .  .  .  .  .  .
2   .  .  .  .  .  .  .
```

```
3    .  .  1  .  .  .  .
4    .  .  1  .  .  1  .
5    .  1  1  .  .  1  .
6    .  1  1  .  .  1  .
7    .  1  1  .  .  1  .
8    .  1  1  .  .  1  .
9    .  1  1  .  .  1  .
10   .  1  .  .  .  1  .
[*,*,2]:    41 42 43 44 45 46 47 :=
0   .  .  .  .  .  .  .
1    .  .  .  .  .  .  .
2    .  .  .  .  .  .  .
3    .  .  .  .  .  .  .
4    .  .  .  1  .  .  .
5    .  .  .  1  1  .  .
6    .  .  .  1  1  .  .
7    .  .  .  1  1  .  .
8    .  .  .  1  1  .  .
9    .  .  .  1  1  .  .
10   .  .  .  1  1  .  .

[*,*,4]:    41 42 43 44 45 46 47 :=
0   .  .  .  .  .  .  .
1    .  .  .  .  .  .  .
2    .  .  .  .  .  .  .
3    .  .  .  1  .  .  .
4    .  .  .  1  .  .  1
5    .  1  .  1  .  .  1
6    .  1  .  1  .  .  1
7    .  1  .  1  .  .  1
8    .  1  .  1  .  .  1
9    .  1  .  1  .  .  1
10   .  1  .  .  .  .  1 ;
param acost :=
                [1] 30000
                [2] 60000
                [3] 40000
                [4] 60000;

param gcost :=
                [1] 300
```

```
              [2] 700
              [3] 500
              [4] 900;


param atsect default 0.0 :=
     [1, *]    41 7.0 43 10.0 44 10.0
     [2, *]    44 10.0 45 10.0
     [3, *]    43 9.0 46 10.0 42 10.0
     [4, *]    44 9.0 47 10.0 42 10.0;

param seccap default 1.0 :=
;
param depcap default 1.0 :=
;
param arrcap default 1.0 :=
;
param cflight default 0.0 :=
[1,2] 1;

param wtime default 0.0:=
[1] 1;

param trt:  41 42 43 44 45 46 47 :=
 1  1  1  1  1  1  1  1
 2  1  1  1  1  1  1  1
 3  1  1  1  1  1  1  1
 4  1  1  1  1  1  1  1;

param etd:=    1 1 2 4 3 3 4 3;
param eta:=    1 3 2 5 3 5 4 5;
param orig:=    1 41   2 44   3 43 4 44;
param dest:=    1 44   2 45   3 42 4 42;

end;
```

C.2 Sample Dataset II

```
data;
param T :=  45;
param F :=  21;
param S :=  44; #40;

set Airports :=        41       42       43       44;

set OD :=  (41,42) (42,43) (43,41) (44,42);

set Sect_od[41,42] := 41 2 1 26 7 6 29 10 15 20 42;
set Sect_od[42,43] := 42 20 15 37 21 16 33 11 34 17 38 12 13 35 14 43;
set Sect_od[43,41] := 43 14 32 8 27 2 41;
set Sect_od[44,42] := 44 18 39 22 21 20 42;

param depcap default 8.0 :=
;
param arrcap default 5.0 :=
;
param G_f default 6.0 :=
; #max ground holding time
param A_f default 6.0 :=
; #max air holding time

param wtime default 1.0 :=
; #max air holding time

param afa := 10; #1.2; #3; #5;
param epsylon := 0.001;
param etd:=    1 1   2 1   3 1   4 1   5 1   6 1   7 1   8 1   9 1   10 1
11 1   12 1   13 1   14 1   15 1   16 1   17 1   18 1   19 1   20 1   21 1;

param eta:=    1 10   2 10   3 9   4 13   5 13   6 13   7 13   8 13   9 13
10 13   11 13   12 13   13 10   14 13   15 10   16 13   17 13   18 13   19 13
20 13   21 13;

param orig:=   1 41   2 41   3 41   4 41   5 41   6 41   7 41   8 41   9 41
10 41   11 41   12 41   13 42   14 42   15 42   16 42   17 42   18 42   19 42
20 42   21 42;
```

```
param dest:=   1 44   2 44   3 43   4 42   5 42   6 42   7 42   8 42   9 42
10 42   11 42   12 42   13 44   14 43   15 44   16 41   17 41   18 41   19 41
  20 41   21 41;



param Sec_Capacity default 5.0 :=
   [*, 15]   3 3   4 3   5 3   6 3   7 3
   [*, 20]   3 3   4 3   5 3   6 3   7 3
   [*, 37]   3 3   4 3   5 3   6 3   7 3
   [*, 16]   8 3   9 3   10 3   11 3   12 3
   [*, 21]   8 3   9 3   10 3   11 3   12 3
   [*, 38]   8 3   9 3   10 3   11 3   12 3
   [*, 17]   13 3   14 3   15 3   16 3   17 3
   [*, 22]   13 3   14 3   15 3   16 3   17 3
   [*, 39]   13 3   14 3   15 3   16 3   17 3
   [*, 18]   18 3   19 3   20 3   21 3   22 3
   [*, 23]   18 3   19 3   20 3   21 3   22 3
   [*, 40]   18 3   19 3   20 3   21 3   22 3
   [*, 19]   23 3   24 3   25 3   26 3   27 3
   [*, 24]   23 3   24 3   25 3   26 3   27 3;

set CFlights :=  (3,7) (1,5) (16,21) (4,16) (14,9);

set next[41,42,41] := 2;
set next[41,42,2] := 1 26 7;
set next[41,42,1] := 6;
set next[41,42,26] := 6;
set next[41,42,7] := 6;
set next[41,42,6] := 29;
set next[41,42,29] := 10;
set next[41,42,10] := 15;
set next[41,42,15] := 20;
set next[41,42,20] := 42;

set next[42,43,42] := 20;
set next[42,43,20] := 15 37 21;
set next[42,43,15] := 33 16;
set next[42,43,37] := 16;
set next[42,43,21] := 16 38;
set next[42,43,16] := 11 34 17;
```

```
set next[42,43,33] := 11;
set next[42,43,11] := 12;
set next[42,43,34] := 12;
set next[42,43,17] := 12 35;
set next[42,43,38] := 17;
set next[42,43,12] := 13;
set next[42,43,13] := 14;
set next[42,43,35] := 13;
set next[42,43,14] := 43;


set next[43,41,43] := 14;
set next[43,41,14] := 32;
set next[43,41,32] := 8;
set next[43,41,8] := 27;
set next[43,41,27] := 2;
set next[43,41,2] := 41;


set next[44,42,44] := 18;
set next[44,42,18] := 39;
set next[44,42,39] := 22;
set next[44,42,22] := 21;
set next[44,42,21] := 20;
set next[44,42,20] := 42;

set previous[41,42,2] := 41;
set previous[41,42,1] := 2;
set previous[41,42,26] := 2;
set previous[41,42,7] := 2;
set previous[41,42,6] := 1 26 7;
set previous[41,42,29] := 6;
set previous[41,42,10] := 29;
set previous[41,42,15] := 10;
set previous[41,42,20] := 15;
set previous[41,42,42] := 20;

set previous[42,43,20] := 42;
set previous[42,43,15] := 20;
set previous[42,43,37] := 20;
set previous[42,43,21] := 20;
```

```
set previous[42,43,16] := 15 37 21;
set previous[42,43,33] := 15;
set previous[42,43,11] := 16 33;
set previous[42,43,34] := 16;
set previous[42,43,17] := 16 38;
set previous[42,43,38] := 21;
set previous[42,43,12] := 11 34 17;
set previous[42,43,13] := 12 35;
set previous[42,43,35] := 17;
set previous[42,43,14] := 13;
set previous[42,43,43] := 14;

set previous[43,41,14] := 43;
set previous[43,41,32] := 14;
set previous[43,41,8] := 32;
set previous[43,41,27] := 8;
set previous[43,41,2] := 27;
set previous[43,41,41] := 2;

set previous[44,42,18] := 44;
set previous[44,42,39] := 18;
set previous[44,42,22] := 39;
set previous[44,42,21] := 22;
set previous[44,42,20] := 21;
set previous[44,42,42] := 20;

param time2sect:=
   [41, *]   0 5    1 3   2 1    3 3    4 5    5 6    6 4   7 3   8 4    9 6
   10 7   11 6   12 5   13 6   14 7    15 9    16 8   17 7   18 8   19 9  20 11
   21 10   22 9   23 10    24 11   25 5    26 3    27 3    28 5    29 6   30 5
   31 5   32 6  33 8   34 7   35 7   36 8    37 10   38 9   39 9   40 10
    41 0.0    42 12   43 8   44 9
   [42, *]   0 9    1 10    2 11    3 12    4 13    5 7    6 8   7 9   8 10
   9 12  10 5   11 6   12 7   13 9   14 11   15 3    16 4   17 6   18 8   19 10
20 1    21 3    22 5   23 7    24 9    25 9    26 10   27 11   28 12   29 7
30 8   31 9   32 11    33 5   34 6   35 8   36 10    37 3    38 5   39 7   40 9
 41 12    42 0    43 12   44 9
   [43, *]   0 11   1 9    2 7    3 6    4 5    5 10   6 8   7 6   8 4    9 3
   10 9   11 7   12 5   13 3   14 1    15 10   16 8   17 6   18 4   19 3   20 11
21 9    22 7   23 6    24 5    25 10   26 8    27 6    28 5    29 9   30 7
31 5   32 3    33 9   34 7   35 5   36 3    37 10   38 8   39 6   40 5
```

```
41 8  42 12   43 0    44 5
   [44, *]   0 10   1 9    2 8    3 7    4 8    5 9    6 7   7 6   8 5    9 6
 10 8   11 6   12 4   13 3   14 4    15 7    16 5   17 3   18 1   19 3    20 8
21 6    22 4   23 3    24 4    25 9    26 8    27 7    28 7    29 8   30 6
31 5   32 5    33 7   34 5   35 3   36 3    37 7    38 5   39 3   40 3
41 9    42 9    43 5    44 0;


param ftime default 2.0 :=
   [2, *] 41  1
   [14, *] 43  1
   [18, *] 44  1
   [20, *] 42  1
   [25, *]   0 1   1 1   5 1   6 1
   [26, *]   1 1   2 1   6 1   7 1
   [27, *]   2 1   3 1   7 1   8 1
   [28, *]   3 1   4 1   8 1   9 1
   [29, *]   5 1   6 1   10 1   11 1
   [30, *]   6 1   7 1   11 1   12 1
   [31, *]   7 1   8 1   12 1   13 1
   [32, *]   8 1   9 1   13 1   14 1
   [33, *]   10 1   11 1   15 1   16 1
   [34, *]   11 1   12 1   16 1   17 1
   [35, *]   12 1   13 1   17 1   18 1
   [36, *]   13 1   14 1   18 1   19 1
   [37, *]   15 1   16 1   20 1   21 1
   [38, *]   16 1   17 1   21 1   22 1
   [39, *]   17 1   18 1   22 1   23 1
   [40, *]   18 1   19 1   23 1   24 1
   [41, *] 2  1
   [42, *] 20  1
   [43, *] 14  1
   [44, *] 18  1;



end;
```

References

A. O. Adewumi. *Some Improved Genetic-Algorithms Based Heuristics for Global Optimization with Innovative Applications.* PhD thesis, University of the Witwatersrand, Johannesburg, South Africa, 2010.

A. O. Adewumi and M. M. Ali. A multi-level genetic algorithm for a multi-stage space allocation problem. *Mathematical and Computer Modelling*, 51(1):109–126, 2010.

A. Agustín, A. Alonso-Ayuso, L. F. Escudero, and C. Pizarro. Mathematical optimization models for air traffic flow management: A review. 2009.

A. Agustín, A. Alonso-Ayuso, L. F. Escudero, and C. Pizarro. On air traffic flow management with rerouting. part i: Deterministic case. *European Journal of Operational Research*, 219(1):156–166, 2012a.

A. Agustín, A. Alonso-Ayuso, L. F. Escudero, and C. Pizarro. On air traffic flow management with rerouting. part ii: Stochastic case. *European Journal of Operational Research*, 219(1):167–177, 2012b.

A. Akgunduz, B. Jaumard, and G. Moeini. Non-time indexed modeling for en-route flight planning with speed-fuel consumption trade-off, 2013.

M. M. Ali, A. O. Adewumi, N. Blamah, and O.Falowo. Mathematical modeling and optimization of industrial problems. *Journal of Applied Mathematics*, 2015(438471): 3pages, 2015.

G. Andreatta and L. Brunetta. Multiairport ground holding problem: A computational evaluation of exact algorithms. *Operations Research*, 46(1):57–64, 1998.

G. Andreatta and G. Romanin-Jacur. Aircraft flow management under congestion. *Transportation Science*, 21(4):249–253, 1987.

G. Andreatta and G. Tidona. A new formulation for the multi-airport ground holding problem. Technical report, Internal Report, 1994.

S. A. ATNS. Air traffic management issue 2 vol 25. *Available at*http://www.worldairops.com/AFI/docs/AFI_SA_ATNSFlowManagement_atWorldAirOps.com.pdf, 2010.

S. A. ATNS. Central airspace management unit atfm road show 2011. *Available at*http://www.atns.co.za/PDF/ProductsandServices/CAMUUpdates/CAMUATFMRoadShow2011.pdf, 2011.

S. A. ATNS. Central airspace management unit atfm status report: South africa. *Available at*http://www.atns.co.za/PDF/ProductsandServices/CAMUUpdates/ATFMStatusReportSouthAfrica.pdf, 2015.

H. Balakrishnan and B. G. Chandran. Optimal large-scale air traffic flow management. *Available at http://www.mit.edu/~hamsa/pubs/BalakrishnanChandran_ATFM.pdf*, 2015. Accessed March 2015.

M. O. Ball, R. Hoffman, A. R. Odoni, and R. Rifkin. A stochastic integer program with dual network structure and its application to the ground-holding problem. *Operations Research*, 51(1):167–171, 2003.

C. Barnhart, D. Bertsimas, C. Caramanis, and D. Fearing. Equitable and efficient coordination in traffic flow management. *Transportation Science*, 46(2):262–280, 2012.

N. Barnier and P. Brisset. Graph coloring for air traffic flow management. *Annals of operations research*, 130(1-4):163–178, 2004.

Z. Beheshti and S. M. Hj Shamsuddin. A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl*, 5(1):1–35, 2013.

D. Bertsimas and S. Gupta. A proposal for network air traffic flow management incorporating fairness and airline collaboration. In *Operations Research*, 2011.

D. Bertsimas and A. R. Odoni. *A critical survey of optimization models for tactical and strategic aspects of air traffic flow management*. National Aeronautics and Space Administration, 1997.

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

D. Bertsimas and S. S. Patterson. The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach. *Transportation Science*, 34(3): 239–255, 2000.

D. Bertsimas, G. Lulli, and A. R. Odoni. The air traffic flow management problem: An integer optimization approach. In *Integer programming and combinatorial optimization*, pages 34–46. Springer, 2008.

D. Bertsimas, G. Lulli, and A.Odoni. An integer optimization approach to large-scale air traffic flow management. *Operations Research*, 59(1):211–227, 2011.

G. Buxi and M. Hansen. Generating probabilistic capacity profiles from weather forecast: A design-of-experiment approach. In *Proc. of USA/Europe Air Traffic Management Research & Development Seminar*, 2011.

S. Chaimatanan. *Strategic planning of aircraft trajectories*. PhD thesis, Université Paul Sabatier-Toulouse III, 2014.

Y. Chang. *Stochastic programming approaches to air traffic flow management under the uncertainty of weather*. PhD thesis, Georgia Institute of Technology, 2010.

J. W. Chinneck. Practical optimization: a gentle introduction. *Available at http://www.sce.carleton.ca/faculty/chinneck/po.html*, 2006. Accessed May 2015.

J. Czyzyk, J. H. Owen, and S. J. Wright. Neos: optimization on the internet. 1997.

D. Delahaye and S. Puechmorel. *Modeling and optimization of air traffic*. John Wiley & Sons, 2013.

P. Dell'Olmo and G. Lulli. A dynamic programming approach for the airport capacity allocation problem. *IMA Journal of Management Mathematics*, 14(3):235–249, 2003.

S. A. Department of Transport. White paper on national civil aviation policy. *Available at http://www.transport.gov.za/Portals/0/Aviation/NewFolder/NCAPdraftfinal3Sept-2015.pdf*, 2015. Accessed November 2015.

L. Epstein, A. Futer, and L. Medvedovsky. Multiple airport scheduling. *Rapport technique, Volpe National Transportation Systems Center, Cambridge, MA, USA*, 1992.

R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A mathematical programming language*. AT&T Bell Laboratories Murray Hill, NJ 07974, 1987.

H. V. Ganu. Modeling air traffic flow management. *arXiv preprint arXiv:0802.0117*, 2008.

E. P. Gilbo. Airport capacity: Representation, estimation, optimization. *Control Systems Technology, IEEE Transactions on*, 1(3):144–154, 1993.

E. P. Gilbo. Optimizing airport capacity utilization in air traffic flow management subject to constraints at arrival and departure fixes. *Control Systems Technology, IEEE Transactions on*, 5(5):490–503, 1997.

S. Gupta and D. Bertsimas. Multistage air traffic flow management under capacity uncertainty: A robust and adaptive optimization approach. 2011.

M. P. Helme. Reducing air traffic delay in a space-time network. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*, pages 236–242. IEEE, 1992.

R. Hoffman, W. Hall, M. Ball, A. Odoni, and M. Wambsganss. Collaborative decision making in air traffic flow management. *Technical Report RR-99-2 NEXTOR, UC Berkeley*, 1999.

R. L. Hoffman. *Integer programming models for ground holding in air traffic flow management*. PhD thesis, National Centre for Excellence in Aviation Operations Research, 1997.

S. Ichoua. A scenario-based approach for the air traffic flow management problem with stochastic capacities. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 7(8), 2013.

K. Kuhn and A. Raith. Biobjective air traffic flow management. *Availbale at https://secure.orsnz.org.nz/conf45/program/Papers/ORSNZ2010_Kuhn.pdf*, 2010. Accessed March 2015.

G. Laporte and I. H. Osman. Routing problems: A bibliography. *Annals of Operations Research*, 61(1):227–262, 1995.

K. S. Lindsay, E. A. Boyd, and R. Burlingame. Traffic flow management modeling with the time assignment model. *Air Traffic Control Quarterly: An International Journal of Engineering and Operations*, 1993.

P. B. Liu, M. Hansen, and A. Mukherjee. Scenario-based air traffic flow management: From theory to practice. *Transportation Research Part B: Methodological*, 42(7):685–702, 2008.

S. Luke. *Essentials of Metaheuristics*. Lulu, 2009. Available for free at http://cs.gmu.edu/$\sim$sean/book/metaheuristics/.

G. Lulli and A. Odoni. The european air traffic flow management problem. *Transportation Science*, 41(4):431–443, 2007.

A. Makhorin. Modeling language gnu mathprog. *Relatório Técnico, Moscow Aviation Institute*, 2015.

J. B. Marron. *The stochastic air traffic flow management rerouting problem*. PhD thesis, Massachusetts Institute of Technology, 2004.

A. Mukherjee and M. Hansen. A dynamic stochastic model for the single airport ground holding problem. *Transportation Science*, 41(4):444–456, 2007.

L. Navazio and G. Romanin-Jacur. The multiple connections multi-airport ground holding problem: Models and algorithms. *Transportation Science*, 32(3):268–276, 1998.

A. R. Odoni. The flow management problem in air traffic control. In *Flow control of congested networks*, pages 269–288. Springer, 1987.

M. O. Olusanya, M. A. Arasomwan, and A. O. Adewumi. Particle swarm optimization algorithm for optimizing assignment of blood in blood banking system. *Computational and mathematical methods in medicine*, 2015, 2015.

I. Oxford Economics. South africa country report: Economic benefits from air transport in south africa. *Available at https://www.iata.org/policy/Documents/Benefits-of-Aviation-South-Africa-2011.pdf*, 2011. Accessed November 2015.

W. Philipp and F. Gainche. Air traffic flow management in europe. In *Advanced Technologies for Air Traffic Flow Management*, pages 64–106. Springer, 1994.

O. Richetta and A. R. Odoni. Solving optimally the static ground-holding policy problem in air traffic control. *Transportation science*, 27(3):228–238, 1993.

O. Richetta and A. R. Odoni. Dynamic solution to the ground-holding problem in air traffic control. *Transportation research part A: Policy and practice*, 28(3):167–185, 1994.

W. Ronald. *Stochastic optimisation models for air traffic flow management.* PhD thesis, Makerere University, 2010.

P. Roosens. Congestion and air transport: a challenging phenomenon. *European Journal of Transport and Infrastructure Research*, 8(2):137–146, 2008.

R. A. Sarker and C. S. Newton. *Optimization modelling: a practical approach.* CRC Press, 2007.

I. Shodhganga. Chapter 4: Optimization techniques in perspective. *Available at http://shodhganga.inflibnet.ac.in/bitstream/10603/11449/9/09_chapter4.pdf*, 2010. Accessed October 2015.

B. Sridhar, S. R. Grabbe, and A. Mukherjee. Modeling and optimization in traffic flow management. *Proceedings of the IEEE*, 96(12):2060–2080, 2008.

M. Terrab and A. R. Odoni. Strategic flow management for air traffic control. *Operations research*, 41(1):138–152, 1993.

W. Tian and M. Hu. Study of air traffic flow management optimization model and algorithm based on multi-objective programming. In *Computer Modeling and Simulation, 2010. ICCMS'10. Second International Conference on*, volume 2, pages 210–214. IEEE, 2010.

S. Torres. Swarm theory applied to air traffic flow management. *Procedia Computer Science*, 12:463–470, 2012.

T. W. M. Vossen, R. Hoffman, and A. Mukherjee. Air traffic flow management. In *Quantitative problem solving methods in the airline industry*, pages 385–453. Springer, 2012.

P. B. M. Vranas, D. Bertsimas, and A. R. Odoni. Dynamic ground-holding policies for a network of airports. *Transportation Science*, 28(4):275–291, 1994.

E. W. Weisstein. Mathworld–a wolfram web resource, 2007. *Available at*http://mathworld.wolfram.com/, 2015.

L. A. Wolsey and G. L. Nemhauser. *Integer and combinatorial optimization.* John Wiley & Sons, 1988.

X. Zhang, Y. Zhou, B. Liu, and Z. Wang. The air traffic flow management with dynamic capacity and co-evolutionary genetic algorithm. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 580–585. IEEE, 2007.