Route generation for optimization in the Air Transport System : aircraft recovery problem and 4D trajectory planning

Qian, Xiongwen

2018

Qian, X. (2018). Route generation for optimization in the Air Transport System : aircraft recovery problem and 4D trajectory planning. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/73437>

<https://doi.org/10.32657/10356/73437>

Route Generation for Optimization in the Air Transport System: Aircraft Recovery Problem and 4D Trajectory Planning

QIAN Xiongwen

School of Mechanical and Aerospace Engineering

A dissertation submitted to Nanyang Technological University in partial
fulfillment of the requirements for the degree of

Doctor of Philosophy

2018

To my family,

“You are the reason I am. You are all my reasons.”

Abstract

The punctuality and efficiency of the Air Transport System (ATS) has a significant impact on the economy. In 2016, the US Passenger Carrier Delay Costs were estimated to be USD 62.55 per minute and the total US flight arrival delays amounted to over 59 million minutes. Furthermore, air traffic volume is growing rapidly. The International Civil Aviation Organization (ICAO) estimates that air traffic in the Asia Pacific region will triple by 2030. With the rapid growth of air traffic, the inefficiency of the ATS has become a salient problem. Flight delays, cancellations, and air traffic congestion cost customers and airlines a considerable amount of money. In this dissertation, we tackle the inefficiency of the ATS from two perspectives: airspace users (airlines) and Air Traffic Management (ATM) service providers (air traffic controllers). Correspondingly, two critical problems are studied on the operational level, the aircraft recovery problem (ARP) and 4D (three dimensions plus time) trajectory planning. Both problems are formulated as special routing problems, and because of the large computational scale of the two problems encountered in real life, they are solved by carefully designed route generation algorithms employing ad hoc decomposition techniques.

In the first part of the dissertation, the aircraft recovery problem with airport capacity constraints and maintenance flexibility is investigated. The problem is to re-schedule flights and re-assign aircraft in real time with minimized recovery cost for airlines after disruptions occur. In most published studies, airport capacity and flexible maintenance are not considered simultaneously via an optimization approach. To bridge this gap, we propose a column generation framework to solve the problem. The framework consists of a master problem for

selecting routes for aircraft and subproblems for generating routes. Airport capacity is explicitly considered in the master problem and swappable planned maintenances can be incorporated in the subproblem. Instead of the discrete delay models that are widely adopted in much of the existing literature, in this work flight delays are continuous and optimized accurately in the subproblems. The continuous-delay model improves the accuracy of the optimized recovery cost. In one test scenario, the accuracy is improved by up to 37.74%. The computational study based on real-world problems shows that the master problem gives a very tight linear relaxation with small, often zero, optimality gap. Large-scale problems can be solved within 5 minutes and the run time could be further shortened by parallelizing subproblems on more powerful hardware. In addition, from a managerial point of view, computational experiments reveal that swapping planned maintenances may bring a considerable reduction in recovery cost from an estimated 20% to 60%, depending on specific problem instances. Furthermore, the decreasing marginal value of airport slot quotas is found by computational experiments.

In the second part of the dissertation, we consider a coordinated multi-aircraft 4D trajectory planning problem which is illustrated by planning 4D trajectories for aircraft traversing an Air Traffic Control (ATC) sector. The planned 4D trajectories need to specify each aircraft's position at any time, ensuring conflict-free trajectories and reducing fuel and delay costs, with possible aircraft maneuvers such as speed adjustments and flight level changes. In contrast to most existing literature, the impact of buffer safety distance is also under consideration, and freedom from conflict is guaranteed at any given time (not only at discrete time instances). The problem is formulated as a pure-strategy game with aircraft as players and all possible 4D trajectories as strategies. An efficient maximum improvement distributed algorithm decomposing multi-aircraft problems into single-aircraft problems is developed to find an equilibrium at which every aircraft cannot unilaterally improve further without the need to enumerate all possible 4D trajectories in advance. Proofs of the existence of the equilibrium and the convergence of the algorithm are given. A case study based on real air traffic data shows that the algorithm

is able to solve 4D trajectories for online applications with an estimated 16.7% reduction in operating costs while allocating an abundant buffer safety distance at the minimum separation point. Computational experiments verify the scalability of the algorithm.

Airlines and ATM service providers are the two vital stakeholders of the ATS; the methodologies and techniques developed in this dissertation make enlightening contributions to improving the operation of the ATS from an optimization point of view. Because decomposition methods are highly problem-dependent, the design of the algorithms requires both domain knowledge and mathematical intuition. However, in general, the proposed route generation algorithms in the dissertation also provide hints to many other real-life large-scale problems in logistics systems, such as railway transportation, maritime transportation, and unmanned aerial vehicle (UAV) transportation.

Acknowledgements

First and foremost, I thank my parents, Qian Rongchang and Yuan Zhaohua, who gave me life and raised me up in spite of all the difficulties. I am also deeply indebted to my fiancée, Cao Yin, her parents, Cao Di and Qian Ming, and her grandmother, Shen Guanbai. I could never have reached so far in my study without their firm support and confidence in me, even during the time I was in doubt of myself. I also thank my uncles and aunties, Yuan Yuehua and He Liqun, Yuan Jianjun and Ma Yuan, and, particularly, Yuan Jianmin and Hong Qin, who are always willing to give me their kind help. This dissertation is dedicated to my beloved grandparents, Qian Hao and Cui Yonglan and Yuan Gensheng and Jiang Shizhen, as well.

I am so fortunate that I have been helped by so many significant professors during my Ph.D. study. I am extremely grateful to my supervisor, Prof Chen Chun-Hsien, for his profound knowledge, patient guidance, and heart-warming encouragement. In addition, my sincere thanks go to my co-supervisor, Prof Chen Songlin, for his insightful suggestion on research and career. I would also express my great gratitude to Prof Duan Fei, for his wisdom and care, which solved my confusion and anxiety in my dark time. What's more, I am privileged to have Prof Wen Changyun and Prof Jitamitra Desai as my thesis advisory committee members, and the discussions with Prof Nie Xiaofeng are always helpful. I will never forget Prof Mao Jianfeng, who brought me to Nanyang Technological University, got me start this marvelous Ph.D. journey, and continued to offer his generous help to me ever after.

Moreover, I would like to extend my special thanks to Prof Liang Zhe, who kindly invited me to do research at Tongji University in Shanghai for one year. It is one year full of my best

memories. I am eternally grateful for the teaching and training that I received during that period of time.

Finally, my biggest thanks to my dear friends I have ever made in Singapore and Shanghai during my Ph.D. life. They are Yang Changpeng, Diao Xudong, Fang Zhou, Tong Yanyan, Sheng Lingling, Ye Zhijian, Yang Songyuan, Hong Jiahua, Zhang Jin, Zheng Meimei, Tang Lihua, Zhang Chuan, Chang Danni, Xu Gangyan, Li Fan, Lee Ching Hung, Lee Yu-Chi, Wang Zuoxu, Li Qiye, Rao Jing, Ai Gang, Li Zhong, Zhang Chaodong, Wang Wenfeng, Gu Jun, Wu Yang, Nguyen Phuong, Kalyan Ghosh, Ashok Krishnan, Nguyen Thi Kim Ngan, Tommy Dam Duc Tai, Gao Fei, Lu Chieh, Ryan Huynh Bao Huy, Koh Hock Seng, Gibson Tan, Ban Haw Leong, John Liew, Qi Xiaofei, Wang Kai, Liu Jianing, Rakesh Prakash, Wang Qingyi, Zhu Tiantian, Chen Qianqian, Zhang Guizhen, Zhang Caizhi, Pan Zehua, Dong Qingbing, Miao Bin, Yi Jun, Xu Haoxin, Zhang Dong, Yu Chuhang, Qin Ying, Wang Mengjie, Zhang Cheng, Xiao Fan, Zhou Lei, Wang Danni, Shang Jieyu, Liang Jingjing, Gao Song, Xing Xin, Zhao Xiaowei, Wei Kai, Xu Hongming, Yang Xuenan, Wang Yilei, Jiang Kaiyan, Zhang Wenhui, Zhang Xiaopeng, Lin Xingkai, Fang Xiufeng, and Jin Xianfei. Forgive me if I missed anyone. If you are not listed on the paper, you are living in my heart.

Table of Contents

Abstract	v
Acknowledgements	ix
List of Figures	xv
List of Tables	xvii
List of Abbreviations	xx
1 Introduction	1
1.1 Motivation	1
1.2 Background	4
1.2.1 Airline Disruptions Management: Handling Regular Irregularities . .	4
1.2.2 Air Traffic Management: Towards 4D Trajectory Based Operations .	7
1.3 Dissertation Structure	10
2 Literature Review	13
2.1 Related Work	13
2.1.1 Airline Service Recovery	13
2.1.2 Aircraft Trajectory Planning	15
2.2 Research Gaps and Objectives	18

Table of Contents

2.2.1	Aircraft Recovery Problem	18
2.2.2	4D Trajectory Planning	21
2.3	Summary	23
3	Solving Aircraft Recovery Problem with Airport Capacity Constraints and Main-	
	tenance Flexibility	25
3.1	Introduction	25
3.2	Contributions of This Work	27
3.3	Problem Description	28
3.3.1	Airports	28
3.3.2	Aircraft	28
3.3.3	Flights	29
3.3.4	Maintenances	29
3.3.5	Routes	31
3.3.6	Disruptions	31
3.3.7	Recovery Options	32
3.3.8	Overall Recovery Cost	34
3.4	Aircraft Recovery Model and Column Generation Framework	34
3.4.1	General Framework	34
3.4.2	Master Problem Formulation	37
3.4.3	Reduced Cost Calculation	39
3.4.4	Solving Subproblems	41
3.4.5	Column Initialization	49
3.5	Computational Study	50
3.5.1	Data Description	50
3.5.2	Computational Results	51
3.5.3	Numerical Experiments for Computational Performance	54

3.5.4	Computational Studies for Operational Insights	59
3.6	Summary	61
4	Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety	
	Distance and Fuel Optimization	63
4.1	Introduction	64
4.2	Contributions of This Work	67
4.3	Discrete Airspace Representation and Problem Formulation	67
4.4	Buffer Safety Distance and Intensity Function	71
4.5	Coordinated Multi-aircraft 4D Trajectory Planning	74
4.5.1	4D Trajectory Planning as a Pure-Strategy Game	74
4.5.2	Maximum Improvement Distributed Algorithm	77
4.5.3	Single-Aircraft 4D Trajectory Optimization	81
4.6	Case Study Based on Real Traffic Data	83
4.7	Computational Experiments for Future Traffic Demand	91
4.8	Summary	95
5	Conclusions and Future Work	97
5.1	Conclusions	97
5.2	Future Work	100
	Publications	103
	References	105
	Appendix A Appendix for Chapter 3	111

List of Figures

1.1	The Air Transport System	2
1.2	A slew of cancellations for British Airways flights at Heathrow Airport	5
1.3	Airline schedule planning process	6
1.4	Components of Air Traffic Management	8
1.5	Typical technical elements of an ATC system	8
1.6	An example 4D trajectory with timed waypoints (SESAR, 2014)	11
2.1	An illustrative example of swapping planned maintenances (a) if swap not allowed (b) if swap allowed	19
2.2	An example of flight copies which are widely used in the existing literature to model flight delays with error	20
3.1	An example of flight swaps (a) before swap (b) after swap	33
3.2	Flow chart of the computational framework for the ARP	36
3.3	An example of a connection network	42
3.4	An example of processing flight-flight arc, suppose Flight 4's dual is zero, turn time is 5 and swap cost is 10 (a) before arc processing (b) after arc processing	45
3.5	An illustration of adaptively delaying flights for airport slots	48
3.6	Comparison of the recovery results with those of the winning teams	53
3.7	IP objective gap between discrete and continuous delay approaches	56

List of Figures

3.8	Run time with number of used threads run on a workstation (IP time is too small to be shown)	58
4.1	Aircraft safety zone	66
4.2	Airway network of Xiamen airspace	68
4.3	Flight levels of one airway	68
4.4	Discretized airways with waypoints and 4D legs	69
4.5	Two aircraft flying two 4D legs	72
4.6	Intensity scale function	73
4.7	A sketch of time-space graph	82
4.8	Xiamen ZSAMAR01 sector	84
4.9	Flight levels and a sample aircraft trajectory	85
4.10	Fuel consumption curves of A320	86
4.11	Impact of intensity weight on 4D trajectories intensity and minimum separation	87
4.12	An encounter of two aircraft (a) recorded in data (b) planned by MIDA	90
4.13	Run time with increasing number of aircraft	93
4.14	Minimum separation with increasing number of aircraft	94

List of Tables

3.1	Characteristics of test scenarios	51
3.2	Cost parameters of test scenarios	51
3.3	Results of test scenarios	52
3.4	Detailed recovery solutions of test scenarios	52
3.5	Comparison of the recovery results with those of the winning teams	53
3.6	Results of experiments on route initialization methods	55
3.7	A comparison of continuous delay and discrete delay approaches	56
3.8	Run time with number of used threads run on a workstation	58
3.9	Scenarios with planned and unplanned maintenances	60
3.10	Cost parameters of Scenario 6 and Scenario 7	60
3.11	Comparison of recovery solutions given swappable and unswappable planned maintenances	60
3.12	Marginal values of airport slot quotas	62
4.1	Results of multi-aircraft 4D trajectory planning under different intensity weights	88
4.2	Estimated gaps to the global optimal solutions	91
4.3	Results of computational experiments with increasing number of aircraft . . .	92

List of Abbreviations

4D TBO	4D Trajectory Based Operations
AOC	Airline Operations Center
ARP	Aircraft Recovery Problem
ARSM	Aircraft Route Selection Model
ATC	Air Traffic Control
ATM	Air Traffic Management
ATS	Air Transport System
CDM	Collaborative Decision Making
CNS	Communication, Navigation, and Surveillance
EUROCONTROL	European Organisation for the Safety of Air Navigation
FAA	Federal Aviation Administration
GRASP	Greedy Randomized Adaptive Search Procedure
IP	Integer Programming
IQP	Integer Quadratic Programming

List of Abbreviations

LP	Linear Programming
MIDA	Maximum Improvement Distributed Algorithm
MILP	Mixed Integer Linear Programming
MRO	Maintenance, Repair, and Overhaul
NextGen	Next Generation Air Transportation System
SESAR	Single European Sky ATM Research
UAV	Unmanned Aerial Vehicle

Chapter 1

Introduction

This chapter presents the motivation and an overview of the entire research of this dissertation. Real-practice contexts are demonstrated for the aircraft recovery problem and 4D trajectory planning. The necessity to develop computational tools to facilitate Airline Disruptions Management is pointed out. Air Traffic Management is briefly introduced with a focus on its evolution towards 4D Trajectory Based Operations.

1.1 Motivation

“Why is my flight delayed?”, or, more dreadfully, “Why is my flight cancelled?” These two simple questions can be among the most expensive ones that we frequently ask in our daily lives. In 2016, the US Passenger Carrier Delay Costs were estimated to be USD 62.55 per minute ([Airlines for America, 2017](#)) and the total US flight arrival delays amounted to over 59 million minutes ([Department of Transportation, 2016](#)). According to an FAA report, the total cost of US air transportation delays was estimated to be USD 32.9 billion in 2007 ([Ball et al., 2010](#)). The punctuality and efficiency of the Air Transport System (ATS) significantly impacts the economy. Unfortunately, how to mitigate flight delay and enhance system efficiency are not easy questions to answer. The planning and operation of a flight involves various stakeholders

Introduction

in the ATS, including the airspace users, airports and Air Traffic Management (ATM) service providers ([Kügler, 2016](#)) (Figure 1.1). System inefficiency might be caused by any of them. In this dissertation, we consider how to improve the operation of the ATS from two perspectives: airspace users (airlines) and ATM service providers (air traffic controllers). Correspondingly, two problems are studied on the operational level of the ATS, the aircraft recovery problem (ARP) and 4D (three dimensions plus time) trajectory planning . Both problems are formulated as special routing problems, and because of the large computational scale of the two problems in real life, they are solved by carefully designed route generation algorithms employing ad hoc decomposition techniques.

Figure 1.1 The Air Transport System ([Kügler, 2016](#))

In the first part of the dissertation, the aircraft recovery problem with airport capacity constraints and maintenance flexibility is investigated. The problem is to re-schedule flights

and re-assign aircraft in real time with minimized recovery cost for airlines after unexpected disruptions happen. In most published studies, airport capacity and flexible maintenance are not considered simultaneously via an optimization approach. To bridge this gap, we propose a column generation framework to solve this problem. The framework consists of a master problem for selecting routes for aircraft and subproblems for generating routes. Airport capacity is explicitly considered in the master problem and swappable planned maintenances can be incorporated into the subproblem. Instead of the discrete delay models that are widely adopted in much of the existing literature, in this work, flight delays are continuous and optimized accurately in the subproblems. Compared to the discrete-delay method, the continuous-delay model improves the accuracy of the optimized recovery cost. In one test scenario, the accuracy is improved by up to 37.74%. The computational study based on real-world problems shows that the master problem gives a very tight linear relaxation with a small, often zero, optimality gap. Large-scale problems can be solved within 5 minutes and the run time could be further shortened by parallelizing subproblems on more powerful hardware. In addition, from a managerial point of view, computational experiments reveal that swapping planned maintenances may bring a considerable reduction in recovery cost from about 20% to 60%, depending on specific problem instances. Furthermore, the decreasing marginal value of airport slot quotas is discovered through the computational experiments.

In the second part of the dissertation, we consider a coordinated multi-aircraft 4D trajectory planning problem which is illustrated by planning 4D trajectories for aircraft traversing an Air Traffic Control (ATC) sector. The planned 4D trajectories need to specify each aircraft's position at any time, ensuring conflict-free trajectories and reducing fuel and delay costs by using allowable aircraft maneuvers such as speed adjustments and flight level changes. In contrast to most of the existing literature, the impact of buffer safety distance is also considered, and freedom from conflicts is guaranteed at any given time (not only at discrete time instances). The problem is formulated as a pure-strategy game with aircraft as players and all possible 4D

Introduction

trajectories as strategies. An efficient maximum improvement distributed algorithm (MIDA) is developed to find an equilibrium at which every aircraft cannot unilaterally improve further without the need to enumerate all possible 4D trajectories in advance. Proofs of the existence of the equilibrium and the convergence of the algorithm are given. A case study based on real air traffic data shows that the algorithm is able to solve 4D trajectories for online use with an estimated 16.7% reduction in monetary costs and allocate even more buffer safety distance at the minimum separation point. The scalability of the algorithm is verified by computational experiments.

As airlines and ATM service providers are the two critical stakeholders of the ATS, the methodologies and techniques developed in this dissertation make enlightening contributions to improving the operation of civil aviation from an optimization point of view. Because decomposition methods are highly problem-dependent, the design of the algorithms requires both domain knowledge and mathematical intuition. However, more generally, we believe that, the proposed route generation algorithms also provide hints to a variety of other real-life large-scale problems in logistics systems, such as railway transportation, maritime transportation, and unmanned aerial vehicle (UAV) transportation.

1.2 Background

1.2.1 Airline Disruptions Management: Handling Regular Irregularities

It was a tough weekend for British Airways. On 27 May 2017, an IT outage knocked out the airline's systems, forced it to cancel nearly 60% of its flights of operation, and stranded about 75,000 passengers over a holiday weekend. The disruption cost the company around GBP 80 million (USD 102.19 million) and it took more than three days for the airline operation to recover ([Dwyer, 2017](#); [Thomson Reuters, 2017a,b](#)).

Figure 1.2 A slew of cancellations for British Airways flights at Heathrow Airport (By Daniel Leal-Olivas/AFP/Getty Images) ([Dwyer, 2017](#))

A disruption can be costly, and, unfortunately, it happens. Besides IT outage, disruptions such as adverse weather, aircraft mechanical problems, and air traffic flow control restrictions are commonly experienced by passengers. Less noticeable causes of disruptions include flight crew absence, sickness, or legal restrictions. Passengers lost in duty-free stores at airports are also a prevalent reason for departure delay. In addition, at airports, there might not be enough ground support equipment because of mechanical failures, which causes disruptions. Security reasons become more and more common for delays in the recent years.

There are many forces that could disrupt the flight schedule. Minor and major irregularities are almost inevitable in airline operations. To some extent, irregularity seems to be regular for airlines. About 20% of US domestic flights arrived late more than 15 minutes, implying some type of schedule disruption, in 2016 ([Department of Transportation, 2016](#)). Among them, weather is the most common one. Furthermore, airlines' hub-and-spoke systems are especially vulnerable to harsh weather at hub airports; flight delays cascade through the system. After a flight schedule is disrupted, a recovery plan needs to be developed and implemented with the objective of returning to the published flight schedule.

Introduction

Airline Disruptions Management is usually coordinated by the Airline Operations Center (AOC) of the airline company. In general, airline schedule planning can be divided into two phases: pre-operational and operational (Figure 1.3). In the pre-operational phase, a flight schedule is developed as the airline's timetable, listing all city-pairs served, flight numbers, departure and arrival times, and aircraft type. About one month before flight operation, aircraft routing and crew scheduling are carried out by the planning department. Then, the flight is handed over to the operating departments and enters its operational phase. The AOC, a centralized group of managers and staff from various departments, coordinates the execution of flights, creating and implementing recovery plans when the flight schedule is disrupted.

Figure 1.3 Airline schedule planning process

In practice, different airlines may have different traditions and philosophies in recovery planning. For example, some airlines are inclined to cancel the day's flights in order to operate normally the next day, while others prefer to operate flights no matter how late they are, assuming that passengers prefer arriving late to having their flights canceled. AOC managers have numerous recovery options to consider singly or jointly when they design a recovery plan, such as flight delay, cancellation, swap, and rerouting. They are discussed in more detail in the

problem description section in Chapter 3. A comprehensive introduction to Airline Disruptions Management in practice can be found in [Cook and Billig \(2017\)](#), from which the background information described in this section largely comes. Designing a recovery plan is a complex task. The number of possible recovery alternatives is enormous, especially for network airlines with high flight frequencies at their hub airports. On the other hand, AOC managers must take various constraints into account, making sure, for example, aircraft are in place for the flights and maintenance requirements are satisfied.

Although assistant support systems are available nowadays for some airlines to help recovery planning, AOC managers still must process a huge amount of information and make decisions largely based on their own experience. Their personal judgments are crucial to a successful recovery. Some airlines even mainly depend on manual work to identify feasible recovery plans. It was reported that Xiamen Airlines, the only National High-quality Prize recipient in the Chinese airline industry, often takes more than 6 hours to finish its recovery planning when major disruptions happen at its hub airports due to adverse weather. It is absolutely necessary to continue to develop advanced computational tools to facilitate Airline Disruptions Management, shorten the recovery process, and save recovery costs.

1.2.2 Air Traffic Management: Towards 4D Trajectory Based Operations

Air Traffic Management consists of Air Traffic Service, Air Space Management and Air Traffic Flow Management ([ICAO, 2007](#)) (Figure 1.4). The main purpose of ATM is to ensure the safety of air traffic, improve efficiency, and avoid air traffic congestion. The rapid growth in air traffic volume brings increasing pressure on the current ATM System. Enabled by the advancement of ATC technologies, 4D Trajectory Based Operations (4D TBO) has been proposed along with other solutions to transform the current system ([FAA, 2013](#)).

Figure 1.4 Components of Air Traffic Management (ICAO, 2007)

As a critical part of Air Traffic Service, Air Traffic Control plays a fundamental role in ATM. A concise but comprehensive introduction of ATC can be found in [Belobaba et al. \(2009\)](#). The main purpose of ATC is to ensure safety by maintaining separations among aircraft. Technically, an ATC system is comprised of three elements, as shown in Figure 1.5. Air traffic controllers observe the air traffic situation using a surveillance system. The controllers issue commands (“clearances”) to aircraft through a communication system and the aircraft fly the cleared route using a navigation system.

Figure 1.5 Typical technical elements of an ATC system ([Belobaba et al., 2009](#))

The increasing demand of air traffic places high pressure on the current ATM system. When air traffic volume approaches or exceeds the capacity of the ATM system, the risk of separation failure among aircraft increases, threatening the safety of air transportation. Initiatives to address the imminent air traffic challenges are underway, such as the Next Generation Air Transportation System (NextGen) plan in the USA ([FAA, 2013](#)) and the Single European Sky ATM Research (SESAR) plan in Europe ([EUROCONTROL, 2012](#)). These two plans both serve as a framework for the long-term future ATM System. NextGen and SESAR share many elements such as implementing satellite-based technologies in navigation and surveillance systems, adopting 4D Trajectory Based Operations, and promoting collaborative decision making processes to improve system efficiency and customer satisfaction.

The first characteristic of the future ATM System is that its development is propelled by satellite-based communication, navigation and surveillance (CNS) technologies. CNS is evolving from ground-based to satellite-based systems. Ultimately, aircraft will be able to receive CNS service anywhere anytime in the sky. Second, empowered by the advancement of satellite-based technologies (e.g., GPS, ADS-B, and CPDLC), as well as improvement in weather forecasting and information sharing, 4D Trajectory Based Operations become a conceivable choice for the future ATM system. 4D TBO primarily focuses on high-altitude cruise operations in en route airspace ([FAA, 2016](#)). Aircraft are expected to be able to fly on precise 4D trajectories which specify both the time and location the aircraft should visit. Airlines will be given more freedom to choose preferred routes to save time and fuel. Controllers and aircraft crews will share the responsibility of ensuring separation among aircraft with the assistance of automatic conflict detection and resolution tools. Safety will be increased. As aircraft current positions and planned 4D trajectories are broadcast among all equipped airspace users, the system is under continuous checks for problems. Delays will be reduced as 4D planning predicts arrival time more accurately. Fuel consumption and emission will decrease, as aircraft will be able to fly more economic trajectories. One initial 4D test flight conducted

by EUROCONTROL in March 2014 demonstrated the feasibility and effectiveness of TBO (EUROCONTROL, 2014). An example of a 4D trajectory with timed waypoints is illustrated in Figure 1.6. In the test, it was found that the flight distance could be reduced by 5%, flight time could be reduced by 2 minutes and fuel burn could be reduced by 12%. 4D TBO is regarded as a central component of the future ATM system. Last but not least, because of the trajectory sharing of 4D TBO, Collaborative Decision Making (CDM) (FAA, 2014) becomes more tangible. Aircraft or airlines may exchange their preferred 4D trajectories and conduct negotiation to resolve conflicts and alleviate airspace congestion before conflicts and congestion actually happen. The future ATM system, capitalizing on new technologies, represents a widespread and transformative change in the management and operation of ATM. Aviation stakeholders are growing more connected, planning in 4D space, and sharing real-time information to bring more efficient and satisfying flights to customers, airlines and aviation authorities.

1.3 Dissertation Structure

The structure of this dissertation is as follows. Chapter 1 presents the motivation and an overview of the entire research work of this dissertation. Background information is provided about the aircraft recovery problem and 4D trajectory planning from real-practice perspectives. In Chapter 2, comprehensive literature reviews of the aircraft recovery problem and 4D trajectory planning are conducted. Research gaps are identified and research objectives are summarized. In Chapter 3, we approach the aircraft recovery problem with airport capacity constraints and maintenance flexibility. A detailed introduction, problem description, the solution, and results are presented. In Chapter 4, we study the 4D trajectory planning while considering buffer safety distance and fuel consumption. A detailed introduction is given, followed by contributions, models, algorithms and computational results. Chapter 5 concludes the dissertation and suggests future work.

Figure 1.6 An example 4D trajectory with timed waypoints ([SESAR, 2014](#))

Chapter 2

Literature Review

In this chapter, comprehensive literature reviews are carried out for the aircraft recovery problem and 4D trajectory planning. Research gaps are identified and research objectives are set.

2.1 Related Work

2.1.1 Airline Service Recovery

There is a rich literature for the airline recovery problem ([Clausen et al., 2010](#)). [Teodorović and Guberinić \(1984\)](#) present one of the pioneering studies in the airline recovery problem. They propose a heuristic which solves each aircraft's route sequentially as a network flow problem using branch-and-bound. [Jarrah et al. \(1993\)](#) develop two network flow models, one for delay and one for cancellation. The models repeatedly solve shortest path problems for necessary flows. [Argüello et al. \(1997\)](#) propose a greedy randomized adaptive search procedure (GRASP) to reconstruct aircraft routes. The algorithm follows a local search paradigm: first an incumbent solution is randomly selected from a candidate list and then neighbors of the incumbent solution are constructed; finally, the most desirable neighbor is put into the candidate list. The local search procedure repeats until a stopping criterion is met. Based on a time-space network

Literature Review

of airports, flight legs and ground arcs, [Yan and Lin \(1997\)](#) devise network flow models that are solved by a network simplex method and a Lagrangian relaxation-based algorithm. [Cao and Kanafani \(1997a,b\)](#) model a quadratic zero-one programming for a multi-fleet recovery problem which is solved by an approximation linear programming (LP) algorithm. Similarly, [Thengvall et al. \(2001\)](#) consider a multi-fleet aircraft recovery problem after hub closures. Three models based on multi-commodity networks are presented. The authors further improve their work in [Thengvall et al. \(2003\)](#), where a bundle algorithm is applied to solving the model. Numerical results show that the bundle algorithm obtains near-optimal solutions much faster than CPLEX, a standard commercial off-the-shelf solver.

Different from network flow based models, [Rosenberger et al. \(2003\)](#) formulate a set partitioning model for rerouting aircraft with a heuristic for pre-selecting the aircraft which are to be rerouted. Selected aircraft are allowed to swap with disrupted aircraft and are included in a route generation procedure. [Andersson and Värbrand \(2004\)](#) develop an approach based on a set packing formulation which is derived from Dantzig-Wolfe decomposition. LP relaxation and a Lagrangian heuristic are proposed for the master problem; two column generation heuristics are implemented for the subproblems. However, maintenance is not considered in their model. [Eggenberg et al. \(2010\)](#) introduce a constraint-specific recovery network for solving the problem. In the network, continuous timeline is discretized into time windows whose width is a parameter that needs to be tuned.

In practice, after solving the aircraft recovery problem, airlines solve the crew and passenger recovery problems to obtain a complete recovery solution. In recent years, with the improvement of modeling approaches and computing capabilities, various methods have been developed to solve the two or three recovery problems in an integrated way. Nevertheless, still, the aircraft recovery problem plays a crucial part in the integrated methodologies. [Petersen et al. \(2012\)](#) publish a fully integrated framework consisting of the three recovery problems using Benders decomposition, which is closely related to the work of [Lettovsky \(1997\)](#). To limit

the size and complexity of the fully integrated problem, only pre-selected flights are input into the model for computation. With respect to the literature solving two recovery problems simultaneously, [Abdelghany et al. \(2008\)](#) utilize a simulation model and resource assignment optimization to solve the aircraft and crew recovery problems. [Maher \(2016\)](#) integrates the crew and aircraft recovery problems by column-and-row generation. Moreover, regarding the joint aircraft and passenger recovery problem, [Bratu and Barnhart \(2006\)](#) propose two passenger recovery models in which passenger itinerary delays and cancellations are estimated in the formulation. A large neighborhood search heuristic is devised by [Bisaillon et al. \(2011\)](#). This heuristic is composed of three phases: construction, repair, and improvement, which iteratively destroys and repairs parts of the solution. Another heuristic-based framework for the joint aircraft and passenger recovery problem is by [Jozefowiez et al. \(2013\)](#). This heuristic also contains several stages; in the first stage, aircraft recovery is performed. Most recently, [Zhang et al. \(2016\)](#) develop a math-heuristic algorithm for recovering aircraft and passengers together. The algorithm carries out an aircraft recovery first; then, flights are re-scheduled and passengers are re-accommodated iteratively until a tolerance limit is reached.

2.1.2 Aircraft Trajectory Planning

Although trajectory planning and vehicle routing have attracted intense attention in the literature ([Delahaye et al., 2014](#); [Laporte, 1992](#)), studies on coordinated multi-aircraft 4D trajectory planning that jointly consider fuel consumption, delay, and buffer safety distance, with flight level change and speed adjustment are limited. By and large, methods for aircraft trajectory planning in the literature can be divided into two categories: mathematical programming and optimal control. Based on discretized time and/or space, mathematical programming models such as integer programming and mixed integer linear programming (MILP) have been formulated to obtain optimized trajectories. [Richards and How \(2002\)](#) describe a method for finding optimal trajectories of multiple aircraft while avoiding collisions. An approximate

Literature Review

model of aircraft dynamics using linear constraints is developed, enabling the MILP approach to be applicable. The model includes multiple waypoint path-planning but requires selected waypoints to be given as input for each aircraft. [Dell'Olmo and Lulli \(2003\)](#) introduce a two-level hierarchical architecture for trajectory optimization for a set of aircraft. On the first level, a flight-by-flight air traffic flow problem is solved for an air route network. Aircraft are specified at what time instances they should be on which arc of the network. The second level carries out optimization for each arc (airway) respectively, solving the detailed positions (coordinates) of the aircraft on the airway, satisfying separation requirements among aircraft. If one airway is congested (no feasible solution is found on the second level), capacity constraints are added to the first level and the problem is resolved. The proper coordination between the two levels of the architecture remains difficult. An aircraft routing problem is defined and proved to be NP-hard by [Roy and Tomlin \(2007\)](#). Integer programming, linear programming relaxation and first-come-first-served heuristics are compared to solve the problem. [Roy and Tomlin \(2007\)](#) assume that all aircraft under consideration fly between the same origin and destination, which is not practical in the real world. [Wei et al. \(2012\)](#) develop a hierarchical decentralized framework for 4D flight planning. Sector capacity is regarded as the main source of constraints; in this framework, no consideration is given to the constraint of proper separation among aircraft. [Campbell \(2012\)](#) publishes a multiscale receding-horizon strategy using MILP that generates environmentally friendly aircraft trajectories. The objective of the work is to minimize fuel consumption, avoiding obstacles such as convective weather areas as hard constraints and contrail formation areas as soft constraints. Taking other aircraft as obstacles, the path-planning algorithm does not consider coordination among multiple aircraft. Similarly, [Zou et al. \(2013\)](#) investigate the optimal 4D aircraft trajectories in a contrail-sensitive environment. The paper first develops a single aircraft trajectory optimization problem as a binary integer program minimizing cost of fuel burn, crew and passenger travel time, CO₂ emissions, and contrail formation; then, it is further extended to planning trajectories for

multiple aircraft on a flight-by-flight basis. Another related work is by [Liang et al. \(2014\)](#) who propose a flight sequence assignment model that decides the levels, tracks, and entry times for flights entering oceanic airspace. Furthermore, [Sherali et al. \(2003\)](#) develop an outstanding work on flight plan selecting problem, which is also related to 4D trajectory planning.

Meanwhile, the optimal control method is also a critical tool for solving 4D trajectories, as surveyed by [Gardi et al. \(2016\)](#). [Prats et al. \(2010\)](#) present a multi-objective optimal control framework for designing single-aircraft departure trajectories that minimize noise annoyance. [Sridhar et al. \(2011\)](#) develop an algorithm to calculate a wind-optimal trajectory for an aircraft in cruise while avoiding regions of airspace that facilitate persistent contrail formation. The problem is formulated as a nonlinear optimal control problem with path constraints, and partial differential equations are derived for the optimal heading. [Kamgarpour et al. \(2011\)](#) formulate aircraft trajectory planning as a hybrid optimal control problem. The aircraft is modeled as a system that switches among three modes. The sequence of modes, switching times, and inputs for each mode are the control variables. An iterative bi-level optimization algorithm is employed to solve the hybrid control problem. [Bonami et al. \(2013\)](#) use a multiphase mixed-integer optimal control approach for solving single-aircraft horizontal flight paths with given waypoints. [Soler et al. \(2014\)](#) extend [Bonami et al. \(2013\)](#) work to 3D space. In their work, step climbs and descents are allowed, and they explored the effect of contrails on single aircraft flight trajectory design by solving a multiphase mixed-integer optimal control problem. [Franco and Rivas \(2015\)](#) also use multiphase optimal control approach to solve single-aircraft trajectory. What is worth noticing is that, since analytical solutions to optimal control problems are often too hard to derive, numerical methods are widely used. Both [Bonami et al. \(2013\)](#) and [Soler et al. \(2014\)](#) convert multiphase mixed-integer optimal control problems to mixed integer non-linear programs and then used a branch-and-bound algorithm to solve them with the help of a commercial solver.

2.2 Research Gaps and Objectives

2.2.1 Aircraft Recovery Problem

Although the ARP has been extensively examined and jointly included in the integrated recovery problems, we observed that some research gaps still need to be further addressed. For the aircraft recovery problem, the first aspect is airport capacity. Airport congestion is a critical cause of delays in the current air transport system. In practice, when adverse weather occurs at an airport, air traffic control may predict that the planned arrivals and departures of the airport will exceed its reduced capacity, so the ATC issues a flow rate control regarding the airport. Each airline is allocated a quota, stipulating the maximum number of departures and arrivals it may have during the flow control period. The airport capacity constraints must be satisfied when the airline devises its recovery plan. However, most ARP studies ([Abdelghany et al., 2008](#); [Andersson and Värbrand, 2004](#); [Argüello, 1997](#); [Argüello et al., 1997](#); [Bierlaire et al., 2007](#); [Bratu and Barnhart, 2006](#); [Cao and Kanafani, 1997a,b](#); [Jarrah et al., 1993](#); [Maher, 2016](#); [Teodorović and Guberinić, 1984](#); [Thengvall et al., 2003, 2001](#); [Yan and Lin, 1997](#)) do not take airport capacity constraints explicitly into account. For the few models incorporating airport capacity constraints ([Bisaillon et al., 2011](#); [Jozefowicz et al., 2013](#); [Rosenberger et al., 2003](#); [Sinclair et al., 2016](#); [Zhang et al., 2016](#)), because of the complexity of the problems, they are usually solved by heuristics which do not guarantee optimality or provide any optimality gap for near-optimal solutions. Although [Petersen et al. \(2012\)](#) provide an optimization approach to airline integrated recovery, it uses a heuristic to preselect flights to be input into its model to reduce model size. In brief, there is indeed a need to develop an optimization approach for the aircraft recovery problem with airport capacity constraints.

The second aspect is aircraft maintenance, which in fact possesses more flexibility than previously thought. Among the limited number of studies considering maintenance in detail ([Bierlaire et al., 2007](#); [Bisaillon et al., 2011](#); [Rosenberger et al., 2003](#); [Zhang et al., 2016](#)),

Figure 2.1 An illustrative example of swapping planned maintenances (a) if swap not allowed (b) if swap allowed

planned regular maintenances are often treated as fixed tasks whose time and space are not allowed to be altered during aircraft recovery. Consequently, after disruptions (e.g., delay by airport closure), the planned maintenances become more difficult to fulfill: an aircraft may have to cancel some flights to catch up with a planned maintenance at its planned fixed airport. The flight cancellations may result in high recovery cost. However, in reality, when an airline plans its maintenance schedule, flexibility is always reserved. For example, if aircraft are required to be maintained every 60 flying hours, airlines typically enforce more stringent requirements such as every 40-45 flying hours ([Barnhart et al., 1998](#)), so that flexibility is available when handling disruption. Furthermore, because many airlines outsource their maintenance to companies specializing in maintenance, repair, and overhaul (MRO), and MRO companies usually provide standard services at more than one airport, aircraft may swap their planned maintenances at different airports. For instance, consider an illustrative example in Figure 2.1. If the blue aircraft's flights are delayed (the dashed arrow), the aircraft cannot catch its planned maintenance unless it cancels its two flights and stays at airport A, waiting to be maintained while it is idle. In contrast, if the blue aircraft is allowed to swap its planned maintenance with the green aircraft, it can be maintained at Airport B and its flights need not be cancelled. The

exploitation of maintenance flexibility may bring considerable savings in recovery cost, as demonstrated in Section 3.5.4.1. However, to the best of our knowledge, little work has been done considering the flexibility of aircraft maintenance.

Figure 2.2 An example of flight copies which are widely used in the existing literature to model flight delays with error

Last but not least, improvements should be further explored in terms of the modeling and solving techniques of the aircraft recovery problem. Firstly, most existing literature adopt a discrete method to model flight delay (Bisaillon et al., 2011; Bratu and Barnhart, 2006; Maher, 2016; Sinclair et al., 2014, 2016; Thengvall et al., 2000, 2003, 2001; Yan and Lin, 1997; Zhang et al., 2016). Flights are copied and separated by predetermined intervals to represent all possible delay options as illustrated in Figure 2.2. The main drawback of this method is that the widths of the intervals are hard to set. As pointed out by Maher (2016), due to the discretization of the delay, the optimized recovery cost is an overestimate of the best possible solution. There potentially exists some better flight within one delay interval but it has to be over-delayed to its following flight copy. It is possible to improve the solution quality by decreasing the interval width. However, this leads to more flight copies, larger optimization models, and dramatically increased computation time. In contrast, modeling delay by continuous time will completely resolve such a dilemma. Secondly, since the ARP is a tactical problem faced by airlines in real-time operation, the ARP needs to be solved as quickly as possible. Solution speed is of

paramount importance to any successful application. Previous studies seldom discuss and verify techniques to further accelerate proposed models and algorithms. Although parallel computing is one promising way to further enhance the computational efficiency of the ARP, as far as we know, it has not been implemented and demonstrated in any previous literature.

In terms of the aforementioned research gaps of the ARP, the research objectives of this study are summarized as follows. First, we aim to provide an optimization framework to solve the aircraft recovery problem with airport capacity constraints. Optimal or near-optimal solutions with optimality gaps can be obtained within a short computation time. Second, maintenance flexibility should be considered in the problem. Managerial insights can be derived for the value of maintenance flexibility by swapping planned maintenances. Third, for the modeling and solving techniques, we try to find better ways to model flight delays more accurately as continuous time rather than discrete options. The ARP can be solved utilizing parallel computing to improve the computational efficiency.

2.2.2 4D Trajectory Planning

The surveyed studies in Section 2.1.2 all shed light on solving the 4D trajectory planning problem, however, because of the various specific problem settings and assumptions, they have their limitations from different perspectives. For instance, [Bonami et al. \(2013\)](#); [Franco and Rivas \(2015\)](#); [Kamgarpour et al. \(2011\)](#); [Prats et al. \(2010\)](#); [Soler et al. \(2014\)](#); [Sridhar et al. \(2011\)](#) focus on single-aircraft trajectory instead of multiple aircraft because of the computational burden resulting from detailed aircraft dynamics required in optimal control. With respect to the literature utilizing mathematical programming, [Roy and Tomlin \(2007\)](#) and [Wei et al. \(2012\)](#) consider sector capacity constraints but not aircraft separation constraints. [Richards and How \(2002\)](#) and [Campbell \(2012\)](#) do not consider aircraft delay. [Zou et al. \(2013\)](#) do not account for speed adjustment. None of the surveyed literature considers the impact of buffer safety distance. In this work, we propose a comprehensive multi-aircraft 4D trajectory

Literature Review

planning framework that considers not only fuel consumption, delay, speed adjustment, and flight level change, but also buffer safety distance to meet practical ATC needs. Furthermore, most literature built on discretized time only ensures conflict-free operation at sampled time instances; however, conflict could still possibly happen between sampled times.

On the other hand, due to the inherent computational hardness (in general, NP-hardness) of solving integer programming, mixed integer programming, or mixed integer nonlinear programming problems, mathematical programming methods find it difficult to solve large-scale problems in real time when too many aircraft are involved. Optimal control faces the same difficulty when it comes to multi-aircraft trajectory planning. Both methods take on a central optimization point of view, neglecting the coordination nature of aircraft when negotiating trajectories in real life. In this work, inspired by the equilibrium theory commonly adopted in many transportation systems ([Byung-Wook, 1995](#); [Chen and Ben-Akiva, 1998](#); [Fisk, 1980](#)), we develop a distributed algorithm framework for coordinated multi-aircraft 4D trajectory planning using a pure-strategy game. The algorithm breaks the problem into subproblems that can be solved efficiently. Its computational workload is scalable with the number of aircraft. Different from traditional traffic assignment in equilibrium-based traffic systems, this algorithm is for online microscopic 4D trajectory planning for individual aircraft, rather than macroscopic traffic flow assignment.

In view of the research gaps identified above, the research objectives are stated as follows. First, we aim to develop models and algorithms for online multi-aircraft 4D trajectory planning. Conflict-free trajectories should be calculated fast enough for online application. Second, due to the safety-criticality nature of ATC, the planned 4D trajectories must ensure conflict-free operation at any time in any place. Possible aircraft maneuvers such as speed adjustments and flight level changes need to be included in the planning. Third, to meet the pragmatic needs of real ATC operations, buffer distances should be reserved in addition to the minimum safety

distance separations among aircraft. Efficient decision support should be provided to allocate buffer safety distance and make rational tradeoffs between operating costs and safety buffers.

2.3 Summary

In this chapter, we reviewed the literature on the aircraft recovery problem and 4D trajectory planning. As two critical problems having a huge impact on the efficiency of air transport, many researcher have studied the two problems. However, there still exist research gaps for further improvement. Consequently, research objectives are set for the following chapters.

Chapter 3

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

In this chapter, the aircraft recovery problem with airport capacity constraints and flexible maintenance is considered. Section 3.1 gives an introduction of the problem. Next, the contributions of this work are summarized in Section 3.2, followed by a detailed problem description in Section 3.3. Section 3.4 discusses the methods developed to solve the problem. The computational study is presented in Section 3.5, and Section 3.6 summarizes this chapter.

3.1 Introduction

As discussed in Chapter 1, disruptions have a large financial impact on the airline industry. In 2016, the US Passenger Carrier Delay Costs were estimated to be \$62.55 per minute ([Airlines for America, 2017](#)) and the total US flight arrival delays amounted to over 59 million minutes ([Department of Transportation, 2016](#)). In addition, disruptions directly influence customers' goodwill to airline companies. For example, in 2007, JetBlue, a US low-cost-carrier, suffered

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

from a severe operational disruption caused by a snow and ice storm. Short of effective and timely recovery solutions, as reported by The Wall Street Journal, "several flights sit for 10 hours, toilets overflowing, nothing to eat but snacks, ironically, cabins so hot and stuffy that doors had to be opened to let fresh air in." ([McCartney, 2007](#)) The company's reputation was substantially damaged. In the end, the majority of management including the airline's founder, David Neeleman, were replaced. It is crucial to develop computational tools for airlines to deal with disruptions and obtain high-quality recovery solutions in real time.

The aircraft recovery problem (ARP), as a fundamental part of Airline Disruptions Management, plays a vital role in every airline's daily operation. Although effective methods ([Barnhart et al., 1998](#); [Gao et al., 2009](#); [Liang and Chaovalitwongse, 2013](#); [Liang et al., 2015](#); [Shao et al., 2017](#)) have been proposed to help produce good pre-operational plans for airlines, when plans are implemented, they are inevitably subject to unpredictable disruptions that force airlines to make modifications in a timely manner. Disruptions may be caused by airline resource shortages such as aircraft mechanical failures or absence of crew members, or they could be due to airport capacity and air traffic control restrictions such as the quota of available airport departure and arrival slots in adverse weather ([Bratu and Barnhart, 2006](#)). When disruptions happen, the Airline Operations Center is responsible for making decisions, re-scheduling airline resources including aircraft, flights and crews, and re-accommodating passengers with the objective of restoring the airline's operation back to the planned schedule with minimized cost. Recovery options often include delaying flights, canceling flights, and changing (swapping) the aircraft for flights. The recovery horizon is generally one to four days.

Because of the vast decision space of the recovery problem and the quick response requirement, in real practice, airlines often decompose the recovery process into several stages and solve them in a sequential manner. Because aircraft are usually airlines' most expensive assets, the aircraft recovery problem is typically solved first. The ARP is to determine for each aircraft when and which flights to operate, with the objective of minimizing the total costs of flight

cancellation, flight delay, and aircraft swap, while satisfying constraints such as maintenance, time and space matches, and airport capacity. Given the determined aircraft routes, in the next stage, a crew recovery problem is solved to re-dispatch crews to aircraft. Finally, passengers are re-accommodated by solving a passenger recovery problem. It should be noted that the aircraft recovery problem is the fundamental stage of the whole recovery process: by and large, if fewer flights are cancelled and delayed in the first stage, fewer crew and passengers need to be re-dispatched and re-accommodated in the second and third stages, respectively.

3.2 Contributions of This Work

We summarize our contributions of this work, which are threefold. First, we propose an optimization approach to solve the aircraft recovery problem with airport capacity constraints. Optimal or near-optimal solutions with rigorous optimality gaps can be obtained within a short computation time. The approach is based on a column generation framework which consists of a master problem selecting routes for aircraft and subproblems generating routes. Airport capacity constraints are explicitly formulated in the master problem. The computational study in Section 3.5.2 reveals that the master problem gives a very tight linear relaxation with a small, often zero, optimality gap. Real-world problems can be solved on a desktop PC within 5 minutes.

Second, operational insights are derived for maintenance flexibility, which is directly considered when solving the subproblems. Maintenances are divided into two groups: planned and unplanned ones. Aircraft, in response to disruptions, are allowed to swap planned maintenances so as to adjust the time and place where the aircraft are maintained. Maintenance requirements including maximum flying hours, maximum number of takeoffs/landings, and maximum intervals between two planned maintenances are directly enforced. The computational study in Section 3.5.4.1 reveals that substantial recovery savings could be achieved by swapping planned maintenances. In addition, by carrying out computational experiments on the slot quota, we

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

precisely quantify the value of airport slot quotas and discover their decreasing marginal effect, which provides managerial guidelines for airlines during the recovery operation.

Finally, regarding the modeling and algorithmic part of this work, two major achievements are made. First, without blindly copying flights to model discretized delays, flight delays are computed accurately in subproblems as continuous variables, thus recovery cost is optimized precisely. Second, to further shorten the computational time, we implement the column generation framework by parallel computing. By parallelizing the subproblems and running them on a more powerful computing machine, we show that the run time is reduced to less than 3 minutes, even for large-scale problems. Theoretically, if subproblems are fully parallelized, the run time could be further decreased by about 50%, i.e., to 1.5 minutes.

3.3 Problem Description

In this section, we give a detailed description of the elements involved in the aircraft recovery problem. Given planned flight schedules, a set of planned and unplanned maintenances, and a set of disruptions, the object of the problem is to determine new flight schedules and aircraft routes during the recovery period with minimized recovery cost.

3.3.1 Airports

Airports play an important role in the ARP because airport capacity must be explicitly considered. To model airport capacity constraints, each airport is associated with some airport slots. A slot is a period of time within which maximum departures and arrivals are specified. Note that airport closures are modeled as airport slots with zero capacity.

3.3.2 Aircraft

An aircraft has the following properties:

1. Start available time: the time after which the aircraft can operate a flight or conduct a maintenance.
2. End available time: the time after which the aircraft must be at an airport without any flight or maintenance assigned to it.
3. Start available airport: the airport where the aircraft is located at its start available time.
4. End available airport: the airport where the aircraft should be located at its end available time.

3.3.3 Flights

A flight requires an aircraft to fly from a departure airport to an arrival airport with a scheduled departure time and arrival time. A flight could be delayed within maximum delay amount. We assume that a flight's duration is fixed. For instance, suppose a flight's scheduled departure time is 7:35 am and its arrival time 9:55 am. If the flight is delayed by 30 minutes, the revised departure time will be 8:05 am and arrival time will be 10:25 am. In addition, each flight is assigned an aircraft to operate the flight, but due to disruption, it might be changed to another aircraft in the recovery solution. A change of the planned aircraft is called a swap.

3.3.4 Maintenances

A maintenance requires an aircraft to stay at an airport from maintenance start time to maintenance end time. There are two kinds of maintenances: planned maintenances and unplanned maintenances.

3.3.4.1 Planned Maintenances

An aircraft's planned maintenances are performed on a regular basis according to the following constraints:

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

1. Maximum flying hours: the maximum accumulated airborne hours allowed between two consecutive planned maintenances.
2. Maximum cycles: the maximum accumulated takeoff/landing times allowed between two consecutive planned maintenances.
3. Maximum interval: the maximum interval allowed between two consecutive planned maintenances.

Aircraft must be maintained before either the maximum flying hours, maximum cycles, or maximum interval, whichever happens first, is reached. After an aircraft's planned maintenance is conducted, its used flying hours, cycles, and days are reset to zero.

Planned maintenances are scheduled by an airline's maintenance department several weeks before the operation. As discussed in Section 3.2, it is widely assumed in the literature that each aircraft can only be maintained at its planned fixed airport within its scheduled start time and end time, and each planned maintenance is exclusively reserved for a particular aircraft. However, in fact, planned maintenances possess flexibility: it is possible to swap and cancel planned maintenances for aircraft as long as each aircraft's maximum flying hours, maximum cycles, and maximum interval are satisfied.

3.3.4.2 Unplanned Maintenances

Unplanned maintenances are mainly caused by unexpected mechanical failure which forces an aircraft to be repaired at the required airport for several hours or days. Airlines also call an unplanned maintenance an Aircraft-on-Ground. Unplanned maintenances are not considered in the pre-operational planning phase by the maintenance department, but they must be carried out according to temporarily added maintenance requirements, which specify the start time and end time of the unplanned maintenances and where they must take place. An aircraft with faulty components must be at the specified airport at the beginning of the maintenance period

and stay there until the maintenance ends. Unplanned maintenances cannot be delayed or swapped. In the model developed in Section 3.4.2, the cancellation of unplanned maintenances entails a huge cost, which indicates the infeasibility of the recovery problem. In reality, if an unplanned maintenance has to be cancelled, it means the maintenance should be re-scheduled in coordination with the maintenance department or the outsourced MRO company.

3.3.5 Routes

An aircraft's route is a sequence of flights and maintenances performed by the aircraft. A route must satisfy the following constraints:

1. Airport match: Any two consecutive tasks (flights or maintenances) in the route should be connected in space. The airport where the first task is completed must be the same as the airport where the second task begins. Note that there are three possible connection types between two consecutive tasks: flight-flight, flight-maintenance and maintenance-flight.
2. Time match: Any two consecutive tasks in the route should also be connected in time. For flight-flight connections, the second flight cannot take off before the first flight's arrival time plus turn time. Turn time is used by airlines to do some preparation (e.g. clean-up) between two flights. For flight-maintenance and maintenance-flight connections, there is no turn time, which means maintenance can begin immediately after an aircraft's arrival, and an aircraft is ready to take off right after maintenance finishes.
3. Aircraft match: Because aircraft are not allowed to swap their unplanned maintenances, an aircraft's route should not contain any unplanned maintenance that does not belong to the aircraft.

3.3.6 Disruptions

Three kinds of disruptions are considered in this work:

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

1. Airport flow control: As introduced in Section 3.2, when airport capacity diminishes, ATC will issue a flow rate control specifying the maximum number of departures and arrivals allowed during a period of time. Airlines may need to revise their original flight schedules and aircraft routes to avoid violating the reduced airport capacity. Note that airport closures are also considered in the airport flow control as airports with zero capacity.
2. Unplanned maintenances: As explained in Section 3.3.4.2, unplanned maintenances are not expected in the original flight schedules and aircraft routes; therefore, they may overlap with some scheduled flights assigned to the aircraft. The disrupted flights might have to be delayed or canceled because of the conflict with the unplanned maintenances.
3. Airport/time mismatches: Because of other various real-world disruptions, the airport match and time match described in Section 3.3.5 can be easily broken with respect to the original plan. For example, because of a volcano eruption in Iceland, hundreds of flights were canceled in Europe (BBC, 2011). As a result, the aircraft originally assigned to the canceled flights could not fly to the arrival airports to operate the next flights, and the airport matches were broken. One feature of our proposed methodology is that it does not require the input aircraft routes and flight schedules to be originally airport and time matched.

3.3.7 Recovery Options

In this work, five recovery options are taken into consideration:

1. Flight delays: Flights are allowed to be delayed, and the flight delay must be less than the maximum delay. The shorter the total delay time, the better the solution quality.
2. Flight swaps: Initially, each scheduled flight is assigned to one aircraft. During the recovery process, the scheduled flight can be alternatively assigned to another aircraft.

Figure 3.1 An example of flight swaps (a) before swap (b) after swap

We call this a flight swap. The fewer the flight swaps, the better the solution quality. Flight swaps are widely used in recovery. For instance, consider the situation demonstrated in Figure 3.1(a). The blue aircraft's first flight is delayed due to airport closure, so it cannot connect with its second flight. In the recovery plan (Figure 3.1(b)), the blue aircraft's second flight is swapped to the green aircraft; and the green aircraft's second flight to the blue aircraft.

3. Flight cancellations: If a flight is not suitable to be assigned to any aircraft, it is cancelled. The cost of flight cancellation is related to passengers' broken itineraries. The fewer total cancellations, the better the solution quality.
4. Maintenance swaps: Only planned maintenances can be swapped among aircraft. Similar to flight swaps, originally, each planned maintenance is scheduled for one particular aircraft. During the recovery process, the planned maintenance can be alternatively

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

assigned to another aircraft. This is called a maintenance swap. Although swaps are permitted, each aircraft must comply with the planned maintenance constraints described in Section 3.3.4.1.

5. Maintenance cancellations: For the flexibility of the model which is developed in Section 3.4.2, planned and unplanned maintenances can be cancelled. The planned maintenance constraints in Section 3.3.4.1 must be still held. On the other hand, if any unplanned maintenance is cancelled, it induces a huge cost, which implies the problem is infeasible.

3.3.8 Overall Recovery Cost

The overall recovery cost is composed of flight delays, flight swaps, and flight cancellations, plus maintenance cancellations and swaps if needed. The aim of the ARP is to determine modified flight schedules and aircraft routes while minimizing the overall recovery cost after disruptions.

3.4 Aircraft Recovery Model and Column Generation Framework

3.4.1 General Framework

In this section, we present the column generation framework used to solve the ARP. Column generation is an effective and efficient methodology which has been used to solve many real-life, large-scale optimization problems where the number of decision variables is too large to enumerate explicitly (Desaulniers et al., 2005; Liang et al., 2014). In this work, to avoid generating all possible aircraft routes, the developed column generation framework for the ARP is composed of two parts, the master problem and the subproblems. In the master problem, a route selection problem is solved. Routes are selected for the aircraft to fly, and the computed

3.4 Aircraft Recovery Model and Column Generation Framework

optimal duals of flights and maintenances are input to the subproblems, guiding the generation of new routes. In subproblems, new routes are generated for aircraft. If routes with negative reduced cost are found, the generated routes are fed to the master problem.

The detailed flow chart of the column generation framework for the ARP is illustrated in Figure 3.2. First, a subset of feasible aircraft routes is initialized by heuristics and provided to the master problem. Then, the master problem is solved and the dual of every constraint is computed. To further improve the solution, a subproblem for each aircraft is solved to generate the route with negative reduced cost given the duals from the master problem. The newly generated routes are then added to the master problem and the updated master problem is resolved. The iteration continues until no route with a negative reduced cost can be found, which implies that the current linear programming solution is optimal. Finally, to obtain an integer solution, the master problem is solved as an integer programming (IP) problem with all aircraft routes that have ever been generated. It is worth noting that aircraft's subproblems are independent of each other, which means that they can be solved in parallel to accelerate the overall computation process. The computational experiments in Section 3.5.2 show that the framework solves real-life problems with small, usually zero, optimality gaps within a short run time.

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

Figure 3.2 Flow chart of the computational framework for the ARP

3.4.2 Master Problem Formulation

The master problem is a linear relaxation of the aircraft route selection model (ARSM), which is defined as follows.

Sets

- C the set of aircraft
- F the set of flights
- M the set of maintenances
- L the set of routes
- S the set of departure/arrival slots

Parameters

- α_f the cost if flight f is canceled
- α_m the cost if maintenance m is cancelled
- $\beta_{c,l}$ the cost if aircraft c flies route l
- $\delta_{f,l} = \begin{cases} 1 & \text{if flight } f \text{ belongs to route } l \\ 0 & \text{otherwise} \end{cases}$
- $\delta_{m,l} = \begin{cases} 1 & \text{if maintenance } m \text{ belongs to route } l \\ 0 & \text{otherwise} \end{cases}$
- u_s the capacity of slot s
- $\phi_{s,l}$ the number of times slot s is used by route l

Decision variables

$$x_{c,l} = \begin{cases} 1 & \text{if aircraft } c \text{ flies route } l \\ 0 & \text{otherwise} \end{cases}$$

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

$$y_f = \begin{cases} 1 & \text{if flight } f \text{ is canceled} \\ 0 & \text{otherwise} \end{cases}$$

$$y_m = \begin{cases} 1 & \text{if maintenance } m \text{ is canceled} \\ 0 & \text{otherwise} \end{cases}$$

Objective function and constraints

$$\min \sum_{f \in F} \alpha_f y_f + \sum_{m \in M} \alpha_m y_m + \sum_{c \in C} \sum_{l \in L} \beta_{c,l} x_{c,l} \quad (3.1)$$

$$\text{s.t.} \quad \sum_{c \in C} \sum_{l \in L} \delta_{f,l} x_{c,l} + y_f = 1, \quad \forall f \in F \quad (3.2)$$

$$\sum_{c \in C} \sum_{l \in L} \delta_{m,l} x_{c,l} + y_m = 1, \quad \forall m \in M \quad (3.3)$$

$$\sum_{l \in L} x_{c,l} \leq 1, \quad \forall c \in C \quad (3.4)$$

$$\sum_{c \in C} \sum_{l \in L} \phi_{s,l} x_{c,l} \leq u_s, \quad \forall s \in S \quad (3.5)$$

$$x_{c,l} \in \{0, 1\}, \quad \forall c \in C, l \in L \quad (3.6)$$

$$y_f \in \{0, 1\}, \quad \forall f \in F \cup M \quad (3.7)$$

The objective function of Equation 3.1 is the sum of the flight cancellation cost, maintenance cancellation cost, and route cost. The route cost includes the costs caused by flight delay, flight swap, and planned maintenance swap, which is detailed in Section 3.4.3. The constraints in Equation 3.2 mean that each flight is either cancelled or covered by one route once. Similarly, the constraints in Equation 3.3 require that each maintenance must be carried out by at most one route. The cancellation of an unplanned maintenance is penalized by a huge cost α_m , which implies the infeasibility of the recovery problem, and the unplanned maintenance must be rescheduled. Although planned maintenances can be cancelled, the subproblem in Section 3.4.4.3 guarantees that the maintenance-related constraints, i.e., maximum flying hours,

3.4 Aircraft Recovery Model and Column Generation Framework

maximum cycles, and maximum interval, are strictly enforced and every aircraft is maintained properly. The constraints in Equation 3.4 ensure that each aircraft at most selects one route to fly. The constraints in Equation 3.5 make sure that every airport slot's capacity is satisfied. Equation 3.6 and Equation 3.7 are the binary constraints for the decision variables. Note that in the master problem, an aircraft's candidate routes are given. Every aircraft selects from its generated candidate routes. Each route contains determined flights and maintenances.

3.4.3 Reduced Cost Calculation

Consider the linear relaxation of the ARSM. Let π_f be the dual variable of the constraint in Equation 3.2 for flight f and π_m be the dual variable of the constraint in Equation 3.3 for maintenance m . Let ρ_c denote the dual variable of the constraint in Equation 3.4 for aircraft c and ω_s be the dual variable of the constraint in Equation 3.5 for slot s . Given aircraft c , the reduced cost $\bar{\beta}_{c,l}$ of its route l is

$$\bar{\beta}_{c,l} = \beta_{c,l} - \sum_{f \in F} \pi_f \delta_{f,l} - \sum_{m \in M} \pi_m \delta_{m,l} - \sum_{s \in S} \omega_s \phi_{s,l} - \rho_c, \quad (3.8)$$

where the route cost $\beta_{c,l}$ of aircraft c flying route l is the summation of all the costs associated with the flights and maintenances comprising the route, i.e.,

$$\beta_{c,l} = \sum_{f \in F} (\beta_{c,f}^{swap} + \beta_{c,f}^{delay}) \delta_{f,l} + \sum_{m \in M} \beta_{c,m}^{swap} \delta_{m,l}. \quad (3.9)$$

In Equation 3.9, the first summation term is the cost incurred by the flights and the second summation term is the cost incurred by the maintenances. Given aircraft c , $\beta_{c,f}^{swap}$ is the swap cost if flight f is not originally assigned to aircraft c ; otherwise, the swap cost is zero. Similarly, $\beta_{c,m}^{swap}$ is the swap cost if maintenance m is not originally planned for aircraft c ; otherwise, the swap cost is zero. Because only planned maintenances can be swapped, for the simplicity of notation, $\beta_{c,m}^{swap}$ is defined to be positive infinity if maintenance m is an

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

unplanned maintenance and it is not originally planned for aircraft c , i.e., aircraft c cannot take an unplanned maintenance m that does not belong to it, which is strictly ensured in the algorithm developed in Section 3.4.4.1. Cost $\beta_{c,f}^{delay}$ is the delay cost of flight f if taken by aircraft c .

Inserting Equation 3.9 into Equation 3.8, after rearrangement, we have

$$\bar{\beta}_{c,l} = \sum_{f \in F} (\beta_{c,f}^{swap} + \beta_{c,f}^{delay} - \pi_f) \delta_{f,l} + \sum_{m \in M} (\beta_{c,m}^{swap} - \pi_m) \delta_{m,l} - \sum_{s \in S} \omega_s \phi_{s,l} - \rho_c. \quad (3.10)$$

The first and second summation terms in Equation 3.10 indicate that the reduced cost $\bar{\beta}_{c,l}$ can be “distributed” among flight f and maintenance m . It implies that, in the subproblem, the cost of including a flight or a maintenance in a route can possibly be priced directly. To further split the third summation term of slots into individual flights, we introduce an indicator $\psi_{f,s}$ which equals 1 if flight f uses slot s and 0 otherwise. Then, $\phi_{s,l} = \sum_{f \in F} \psi_{f,s} \delta_{f,l}$. Substituting this into Equation 3.10, we reach Equation 3.11,

$$\bar{\beta}_{c,l} = \sum_{f \in F} (\beta_{c,f}^{swap} + \beta_{c,f}^{delay} - \pi_f - \sum_{s \in S} \omega_s \psi_{f,s}) \delta_{f,l} + \sum_{m \in M} (\beta_{c,m}^{swap} - \pi_m) \delta_{m,l} - \rho_c, \quad (3.11)$$

after rearranging.

Equation 3.11 shows that the reduced cost is contributed by flights, maintenances, and the dual variable of aircraft c . The subproblem is to find a route l for aircraft c with negative reduced cost, i.e., $\min_l \bar{\beta}_{c,l} < 0$. Denote $\bar{\beta}_{c,f} \triangleq \beta_{c,f}^{swap} + \beta_{c,f}^{delay} - \pi_f - \sum_{s \in S} \omega_s \psi_{f,s}$ as the in-effect cost of flight f in the subproblem. The in-effect flight cost $\bar{\beta}_{c,f}$ depends not only on the flight swap cost and delay cost, but also on the duals of the flight and its used slots. The additional two terms of duals can be explained intuitively. For example, according to Equation 3.2, π_f 's sign is unrestricted. During the iteration of the column generation process, after solving the master problem, if $\pi_f > 0$, it implies that flight f is not popularly covered by the aircraft's routes in the master problem, so π_f is deducted from the flight's in-effect cost to encourage more aircraft

3.4 Aircraft Recovery Model and Column Generation Framework

to use the flight in the subproblems. On the other hand, we denote $\bar{\beta}_{c,m} \triangleq \beta_{c,m}^{swap} - \pi_m$ as the in-effect cost of maintenance m . This is an adjusted maintenance cost used in the subproblem with the same spirit as the in-effect flight cost. Substituting the in-effect flight cost $\bar{\beta}_{c,f}$ and maintenance cost $\bar{\beta}_{c,m}$ in Equation 3.11, the reduced cost of aircraft c 's route l becomes

$$\bar{\beta}_{c,l} = \sum_{f \in F} \bar{\beta}_{c,f} \delta_{f,l} + \sum_{m \in M} \bar{\beta}_{c,m} \delta_{m,l} - \rho_c. \quad (3.12)$$

Since ρ_c is a constant given by the master problem, the subproblem for every aircraft c becomes an optimization problem of generating a route l that minimizes its total in-effect cost caused by the flights and maintenances selected in the route.

3.4.4 Solving Subproblems

The goal of the subproblems is to find routes with negative reduced cost. In each iteration of the column generation framework, $|C|$ subproblems are solved. For each subproblem of one aircraft, the optimal route with the most negative reduced cost is obtained. The process is repeated until there is no route with a negative reduced cost found for any aircraft, which indicates that the current LP solution of the ARSM is optimal. In general, we formulate the subproblem as a special shortest path problem on a connection network. For the clarity of presentation, in Section 3.4.4.1, the basic formulation of the subproblem is presented first without considering airport slots and swappable planned maintenances. Then, in Section 3.4.4.2 and Section 3.4.4.3, these two advanced features are added.

3.4.4.1 Basic Formulation: A Multi-label Algorithm

The basic formulation of the subproblem does not consider airport slots and swappable planned maintenances, but it lays a solid foundation for more complicated problem settings. First, for the subproblems, we build a connection network $G(V, A)$ according to the given schedules of

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

flights and maintenances and their operation airports. In the network, nodes V are flights and maintenances, and arcs A are the connections among them. Note that maintenances are treated as a kind of special flight connecting to and from the same airport, so we focus more on flights in the following discussion. Two nodes are directly connected if the airport where the first flight is completed is the same as the airport where the second flight begins. Let earlier departure flights point to later ones. To facilitate finding a route for one aircraft in one subproblem, dummy source and sink nodes are added to the network for the start available airport and end available airport of the aircraft. An example of a flight connection network is illustrated in Figure 3.3, where the aircraft under consideration is scheduled to fly from airport A to airport D. The two numbers in the bracket under each flight node are the flight's scheduled departure time and arrival time converted in integers, respectively. Note that Node 2 stands for a maintenance conducted at airport B. Each flight's or maintenance's in-effect cost $\bar{\beta}_{c,f}$ or $\bar{\beta}_{c,m}$ is associated with the arc pointing to the node.

Figure 3.3 An example of a connection network (The two numbers in a bracket are one flight's scheduled departure time and arrival time converted in integers)

In the subproblem, the major decision variables are $\delta_{f,l}$ and $\delta_{m,l}$, i.e., whether flight f and maintenance m are included in route l . The subproblem becomes one of finding the shortest path from the source node to the sink node in the connection network to minimize the reduced cost, which is equivalent to the length of the shortest path found. However, the tricky part is that a flight's in-effect cost $\bar{\beta}_{c,f} = \beta_{c,f}^{swap} + \beta_{c,f}^{delay} - \pi_f$ is not fully given when the connection network

3.4 Aircraft Recovery Model and Column Generation Framework

is built because the flight's delay cost $\beta_{c,f}^{delay}$ is also a decision variable in the subproblem. We need to find the shortest path on a network where the traveling cost on the arcs are not completely known. The good news is that it turns out the connection network is an acyclic directed graph since time is not reversible. We propose a multi-label (two-label) algorithm in a dynamic programming framework to solve the subproblem. A label has two elements, path cost and delay, because the shortest path's cost from the source node to any flight node i depends not only on the shortest path cost to node i 's predecessor node, but also on node (flight) i 's own delay, and how much flight i should be delayed depends on the delay of its predecessor. Flight delays are decided along arc processing by taking advantage of the fact that the shorter the flight's delay the better.

Denote label b_i of node i as $\langle \bar{\beta}_i, \beta_i^d \rangle$, where $\bar{\beta}_i$ records the path cost from the source node to node i and β_i^d is the delay cost of the flight represented by node i . Equivalently, label b_i of node i stands for a partial path (route) from the source node to node i . Moreover, suppose two labels b_i^1 and b_i^2 are in node i 's label set B_i , i.e., the two labels represent two different paths from the source node to node i . We define that label b_i^1 dominates label b_i^2 if and only if (1) $\bar{\beta}_i^1 \leq \bar{\beta}_i^2$ and (2) $\beta_i^{d,1} \leq \beta_i^{d,2}$. The basic multi-label shortest path algorithm for the subproblem is summarized in Algorithm 3.1.

The algorithm begins by initializing each node's label set as $\emptyset$ except for the label set of the source node. For the source node, its label set is initialized as $\{\langle 0, 0 \rangle\}$ and it is treated as a maintenance with scheduled end time equal to aircraft c 's start available time. The main loop checks each node in topological order, which means that when a node is checked, all its predecessors have been checked. When a node is checked, the arcs linking the node to its successors are processed one by one according to the connection types to which the arcs belong (flight-flight, flight-maintenance, and maintenance-flight). For a flight-flight arc, given one label (partial path and delay) of the first flight, it is computed how much the second flight needs to be delayed right after the first flight. Note that here, in contrast to many previously

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

Algorithm 3.1 Basic multi-label shortest path algorithm

Input: aircraft c and its dual ρ_c , flight dual π_f , maintenance dual π_m , graph $G(V, A)$

Output: route l with the minimum negative reduced cost if any

Initialize the source node's label set as $\{(0, 0)\}$ and the other nodes' label sets as empty, i.e., $B_i = \emptyset, \forall i \in F \cup M$.

```
for each node  $i$  of network  $G$  in topological order do
  for each node  $j$  that is a successor of node  $i$  do
    if node  $i$  is a flight and node  $j$  is a flight then
      Process flight-flight arc  $(i, j)$ .
    end if
    if node  $i$  is a flight and node  $j$  is a maintenance then
      Process flight-maintenance arc  $(i, j)$ .
    end if
    if node  $i$  is a maintenance and node  $j$  is a flight then
      Process maintenance-flight arc  $(i, j)$ .
    end if
  end for
end for
Select the label  $b_i^*$  with the minimum path cost  $\bar{\beta}_i^*$  from the labels of the nodes connecting to the sink node.
if  $\bar{\beta}_i^* - \rho_c < 0$  then
  Construct route  $l$  by repeatedly tracing back the predecessors of  $b_i^*$ .
  return  $l$ .
else
  return null.
end if
```

3.4 Aircraft Recovery Model and Column Generation Framework

Figure 3.4 An example of processing flight-flight arc, suppose Flight 4's dual is zero, turn time is 5 and swap cost is 10 (a) before arc processing (b) after arc processing

published works (Bisaillon et al., 2011; Bratu and Barnhart, 2006; Maher, 2016; Thengvall et al., 2003, 2001; Yan and Lin, 1997; Zhang et al., 2016) that duplicate flights to model flight delays blindly and roughly, flight delay is calculated adaptively and precisely in this process. Next, the computed delay is checked against the maximum delay allowed and the aircraft's end available time; the path cost to the second flight node is computed considering whether an aircraft swap takes place. Finally, if the newly generated label with the computed path cost and flight delay is not dominated by any existing label in the second flight node's label set, it is inserted and the second flight's label set is updated. One example of processing a flight-flight arc is illustrated in Figure 3.4, in which we suppose that Flight 4's dual is zero, turn time is 5, and swap cost is 10.

Alternatively, for a flight-maintenance arc, given one label (partial path and delay) of the flight node, the maintenance is first checked to determine whether it is scheduled for the aircraft under consideration. If not, because it is assumed in this section that maintenance cannot be swapped, the flight-maintenance arc processing ends, so it is ensured that the generated route in the subproblem does not contain any maintenance that does not belong to the aircraft. Furthermore, because maintenance cannot be delayed, if it is detected that the flight arrives after the maintenance begins, the current label is skipped and the next label of the flight node begins

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

processing. There is no turn time required between flight and maintenance. The rest of the process is almost the same as the flight-flight arc processing. Similarly, for a maintenance-flight arc, its processing is almost identical to the flight-flight arc, except that the maintenance's delay must be zero and no turn time is needed.

After all arcs have been processed, the algorithm selects the label b_i^* with the minimum path cost $\bar{\beta}_i^*$ from the labels of the nodes connecting to the sink node. If the obtained reduced cost $\bar{\beta}_{c,l} = \bar{\beta}_i^* - \rho_c < 0$, aircraft c 's route l is constructed by tracing back the predecessors of b_i^* and returned to the master problem.

Next, for the completeness of algorithm design, we prove that Algorithm 3.1 finds the optimal route with the minimum reduced cost for the subproblem. First, we start with a lemma.

Lemma 3.1. *If one label (i.e., a partial route) is a part of an optimal route, any label dominating it in the same label set can also be a part of an optimal route.*

Proof. Suppose label $b^1 = \langle \bar{\beta}^1, \beta^{d,1} \rangle$, $b^2 = \langle \bar{\beta}^2, \beta^{d,2} \rangle$. They are in the same label set of the same node (flight). Labels b^1 and b^2 stand for two different partial routes, l^1 and l^2 , respectively, from the source to the node. Suppose b^1 dominates b^2 , and b^2 (i.e., partial route l^2) is a part of an optimal route. We can build a new route that is composed of two parts: the first part is partial route l^1 and the second part is the remaining route in the optimal route excluding l^2 . Because b^1 dominates b^2 , i.e., route cost $\bar{\beta}^1 \leq \bar{\beta}^2$ and flight delay $\beta^{d,1} \leq \beta^{d,2}$, the new route must be feasible and no worse than the optimal route. The new route is also an optimal route. Label b^1 (i.e., partial route l^1) can also be a part of an optimal route. $\square$

Due to Lemma 3.1, dominated labels are not inserted into the label sets of flight nodes. Because (1) if the dominated label (partial route) is a part of the optimal route, the label dominating it in the label set can replace it as a part of the optimal route and (2) if the dominated label (partial route) is not a part of the optimal route, it should be discarded and hence not inserted into the label set.

We now consider the optimality of the route found by Algorithm 3.1.

Theorem 3.2. *Algorithm 3.1 finds the optimal route with the minimum reduced cost for the subproblem.*

Proof. We prove the theorem by induction. First, obviously, the optimal route from the source to itself is represented by its initialized label $\langle 0, 0 \rangle$, which means zero route cost and zero delay. Then, consider a node and its predecessors in the connection network. Each predecessor has its label set recording all possible non-dominated labels (partial routes) from the source to the predecessor. Because any route (label) from the source to the node under consideration must be extended from one label of its predecessor and all the non-dominated labels of its predecessors are enumerated, the extended label with the minimal path cost in the node's label set represents the optimal route to the node under consideration. $\square$

3.4.4.2 Subproblem with Airport Slots

As mentioned in Section 3.2, airport capacity is an indispensable part of the ARP, and it is modeled as airport slots which specify the maximum number of departures and arrivals within specific periods of time. To explicitly take airport slots into account in the subproblem, the term $\sum_{s \in S} \omega_s \psi_{f,s}$ should be included in the in-effect cost $\bar{\beta}_{c,f}$ of flight f if taken by aircraft c . However, similar to the issue of deciding flight delay in the subproblem, $\psi_{f,s}$, i.e., whether or not flight f uses slot s , is also a decision variable that renders the traveling cost on the arcs unknown in the connection network. Furthermore, to make things worse, a shorter flight delay is no longer necessarily better: flights might be delayed longer for favorable slots in return. To overcome these difficulties, we propose to adaptively delay flights according to airport slots during the arc processing steps of Algorithm 3.1 so that the possible slots that one flight might fall in and possible duals that one flight can have are fully covered. After a flight is delayed by its preceding flight or maintenance, it is further delayed to the points where slots begin and end. For example, consider Flight 1 in Figure 3.5, originally it falls within no slot, so its in-effect cost is $\bar{\beta}_{c,f} = \beta_{c,f}^{swap} + \beta_{c,f}^{delay} - \pi_f$, which does not include any slot dual. Then, it is delayed to the

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

point when it enters the departure slot, because at this point, its in-effect cost $\bar{\beta}_{c,f}$ is changed by deducting the departure slot's dual ω_{dep} , i.e., $\bar{\beta}_{c,f} = \beta_{c,f}^{swap} + \beta_{c,f}^{delay} - \pi_f - \omega_{dep}$. Accordingly, a new label is created by the delay and the in-effect cost. If the label is not dominated by any existing label in the label set of Flight 1, it is inserted, which means a new non-dominated route is found from the source node to Flight 1; otherwise, the route is discarded for further consideration. The process iterates for every point whenever Flight 1 enters or leaves a slot. The pseudo-code of the process of adaptively delaying flights according to airport slots is given in Algorithm 3.2.

Figure 3.5 An illustration of adaptively delaying flights for airport slots

Algorithm 3.2 Adaptively delaying flights for airport slots

Input: node (flight) i , the slots of the departure and arrival airports of flight i

Output: node i 's label set B_i

if flight i does not fall in any slot **then**

 Create label b_i with no dual of any slot.

else

 Create label b_i by the duals of the slots which flight i falls in.

end if

if b_i is not dominated by any existing label in B_i **then**

 Insert b_i into label set B_i .

end if

for each slot behind flight i **do**

for each time point where the slot begins or ends **do**

 Delay flight i by β_i^d to the time point.

 Create label b_i by β_i^d with the duals of the slots that flight i falls in.

if b_i is not dominated by any existing label in B_i **then**

 Insert b_i into label set B_i .

end if

end for

end for

return label set B_i .

3.4.4.3 Subproblem with Swappable Planned Maintenances

The second enrichment of the subproblem is to swap planned maintenances. At first glance, it seems that this task is not that difficult because maintenances are treated as special flights in the connection network and flights can be swapped. However, unfortunately, because aircraft's planned maintenances are not synchronized, after swapping, the maximum flying hours, maximum cycles, and maximum interval constraints of an aircraft's planned maintenance might no longer be feasible. Thus, the multi-label shortest path algorithm, Algorithm 3.1, must be further modified. First, three new label elements, u_i^h , u_i^o , and u_i^v are added to one label, i.e., a label b_i of node i is redefined as $\langle \bar{\beta}_i, \beta_i^d, u_i^h, u_i^o, u_i^v \rangle$ where u_i^h is the used flying hours, u_i^o is the used number of takeoffs, and u_i^v is the elapsed time since the last planned maintenance along the partial route represented by flight i 's label b_i . Second, three new conditions are added to the label dominant rule. Label b_i^1 dominates label b_i^2 if and only if (1) $\bar{\beta}_i^1 \leq \bar{\beta}_i^2$, (2) $\beta_i^{d,1} \leq \beta_i^{d,2}$, (3) $u_i^{h,1} \leq u_i^{h,2}$, (4) $u_i^{o,1} \leq u_i^{o,2}$, and (5) $u_i^{v,1} \leq u_i^{v,2}$. Third, in the flight-maintenance arc processing, a swap cost is incurred if it is a planned maintenance but not for the aircraft c under consideration. Besides, u_i^h , u_i^o , and u_i^v are reset to zero whenever a planned maintenance node is visited. Last but not least, at the beginning of arc processing, a check is added to determine whether there are enough remaining flying hours, number of takeoffs, and interval time to fly the following flight j . If there are, u_i^h , u_i^o , and u_i^v are updated and used to create a new label to be inserted into the label set of node j if the new label is not dominated. The final algorithms for the subproblem with swappable planned maintenances are detailed in Appendix A.

3.4.5 Column Initialization

Column initialization methods can have a significant impact on the run time of the column generation framework. Initial routes are guesses of the optimal ones. If the guesses are 100% correct, optimal solutions are obtained immediately. In this work, two column initialization methods are proposed to generate the initial routes of aircraft for the master problem: initializa-

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

tion by originally feasible planned routes and by a shortest path heuristic. On the one hand, given the recovery character of the ARP, optimal solutions usually resemble the original routes planned for aircraft. Hence each aircraft's planned route is used for initialization if it remains feasible after disruptions. On the other hand, assuming flight duals are zero and considering only swap cost and delay cost, subproblems are solved to initialize aircraft's routes as shortest path problems. These routes represent the preferred routes of an aircraft, regardless of other aircraft's choices. Note that one subproblem can output more than one initial route, if multiple non-dominant routes can be found in the sink node. The numerical experiments in Section [3.5.3.1](#) corroborate the effectiveness of using the two initialization methods together.

3.5 Computational Study

3.5.1 Data Description

To validate the performance of the proposed column generation framework, a computational study was carried out and the results are presented in this section. Five scenarios from real airline operation data were tested. The five scenarios were used as benchmark problems for the Airline Operations Research Competition organized by [Sabre Airline Solutions \(2016\)](#). The characteristics of the scenarios are summarized in Table [3.1](#). The scenarios cover a wide range of disruption types and problem scales. Scenarios 1 and 2 are small scenarios with less than 100 flights. Scenario 3 is medium-sized. Scenarios 4 and 5 are large-scale ones. Because the competition does not differentiate planned and unplanned maintenances, all maintenances are assumed to be unplanned and hence not swappable. The impact of swapping planned maintenances is investigated in Section [3.5.4.1](#).

The cost parameters, i.e., swap cost per flight, delay cost per minute, and cancellation cost per flight, are given by airline companies. In real practice, those cost parameters are determined by airlines' own preferences and open to adjustment according to the airlines' needs. Different

Table 3.1 Characteristics of test scenarios

Scenario	No. of flights	No. of aircraft	No. of maint.	No. of airports	Duration (h)	Disruption
1	59	16	23	12	80	Time / Airport mismatch
2	95	12	0	25	48	Airport closure of 4 hours
3	417	85	25	35	44	Time / Airport mismatch ATC flow control of 1 hour
4	586	44	24	37	85	Aircraft on ground for 48 hours
5	638	44	29	32	90	Aircraft on ground for 8 hours

airline companies have different focuses on their cost components. The cost parameters of the five test scenarios are given in Table 3.2. Moreover, for all scenarios, the turn time between two flights is 30 minutes, and the maximum flight delay allowed is 180 minutes, even though more complicated settings can be accommodated such as a conditional turn time depending on flight types (e.g., domestic or international) in this work.

Table 3.2 Cost parameters of test scenarios

Scenario	Cost parameters		
	Swap per flight	Flight delay per minute	Cancellation per flight
1	10	4	500
2	1	10	800
3	1	8	500
4	10	4	500
5	10	4	500

3.5.2 Computational Results

The column generation framework was programmed in C++ and run on a desktop computer with a 3.9 GHz Intel i3 CPU and a Windows 10 operating system. The LP and final IP of the master problem were solved by the commercial solver CPLEX 12.7. Computational results are listed in Table 3.3, where “LP obj.” is the final LP objective value of the master problem when the column generation iteration terminates and “IP obj.” is the final IP objective value. Moreover,

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

“#CG iterations” is the total number of column generation iterations and “#Routes” stands for the total number of routes generated. The subproblems were implemented by multi-thread programming. Hence, the subproblems were divided to threads and the subproblems in different threads were solved in parallel. Four threads, the maximum number of threads supported by the computer, were used for running the program. Taking Scenarios 4 and 5 for example, there are 44 aircraft and hence 44 subproblems per column generation iteration. Therefore, each thread was responsible for solving $44/4 = 11$ subproblems in parallel. Table 3.4 summarizes the detailed recovery solutions of the scenarios, and Table 3.5 compares the recovery cost and runtime with those of the winning teams in the competition. Tongji-TOBU solved the problem by randomly selecting from enumerated routes and putting selected promising routes into an IP for optimization (Zhang, 2017). SJTU-THR proposed a local search paradigm which is similar to GRASP (Argüello et al., 1997). Eindhoven Univ. of Tech-OPAC developed math-heuristics to generate routes and select from them.

Table 3.3 Results of test scenarios

Scen- ario	LP obj.	IP obj.	Optimality gap	#CG iterations	#Routes	Run time (s)
1	1010.00	1010	0.00%	8	194	0.37
2	4364.00	4364	0.00%	26	396	0.73
3	4781.00	4781	0.00%	75	6,166	19.72
4	2124.29	2138	0.65%	262	11,455	145.84
5	218.00	218	0.00%	351	14,972	251.33

Table 3.4 Detailed recovery solutions of test scenarios

Scen- ario	No. of flight cancellations	No. of flight swaps	Total flight delay (min)	Recovery cost
1	2	1	0	1010
2	2	14	275	4364
3	4	21	345	4781
4	2	89	62	2138
5	0	21	2	218

Table 3.5 Comparison of the recovery results with those of the winning teams

Scenario	KPIs	Tongji-TOBU	SJTU-THR	Eindhoven Univ. of Tech-OPAC	This work
1	Recovery cost	1010	1010	1010	1010
	Run time (s)	10.1	1620	10.2	0.37
2	Recovery cost	4364	4364	4390	4364
	Run time (s)	10.2	1500	10.3	0.73
3	Recovery cost	4781	4781	4848	4781
	Run time (s)	105	1620	10.1	19.72
4	Recovery cost	2730	2434	5200	2138
	Run time (s)	139.8	1740	720	145.84
5	Recovery cost	230	410	1040	218
	Run time (s)	150	1680	720	251.33

Figure 3.6 Comparison of the recovery results with those of the winning teams

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

The results of the five test scenarios verify the effectiveness and efficiency of the column generation framework. As indicated in Table 3.3, for the two small scenarios, the problems were solved almost instantaneously. For the middle-sized problem, it was solved within 1 minute. Moreover, for the two large-scale problems, they were solved in less than 5 minutes. In addition, along with the short run time, solution quality was not compromised. Because the master problem is a linear relaxation of the ARSM, which is an IP, the LP obj. column provides a lower bound of the true optimal objective. Given the integer solution obtained (the IP obj. column), the optimality gap is computed by $(\text{IP obj.} - \text{LP obj.})/\text{LP obj.}$. Of the five test scenarios, four scenarios were proved to be optimal by their zero optimality gaps, except for Scenario 4, which yields a very small gap. The good quality of the solutions is also confirmed by the comparison with the results obtained by other methods in Table 3.5 and Figure 3.6. Better solutions are achieved for large-scale scenarios by this work within short run time; for the small- and medium-size scenarios, the optimal solutions are found much faster. The column generation framework proposed in this study displays clear superiority over the methods proposed by others. Furthermore, in view of the detailed recovery solutions shown in Table 3.4, it is observed that, due to the relatively high cancellation cost, flights are rarely cancelled in the recovery solutions. Moreover, it is obvious that Scenarios 2 and 3 require much more flight delay than other scenarios because disruptions such as airport closure and ATC flow control on hub airports have a more severe influence on flight operation than other kinds of disruptions.

3.5.3 Numerical Experiments for Computational Performance

3.5.3.1 Impact of Column Initialization

As discussed in Section 3.4.5, the initial columns, i.e., initial aircraft routes, play a critical role in the computational performance of the column generation framework. In this section, taking the two large-scale scenarios, Scenarios 4 and 5, we experimented on the two initialization

methods, namely, the feasible planned routes and shortest path heuristic, and check their impact on run time. In addition, the situation with no initial column was tested as a benchmark for comparison. The computational settings were the same as those in Section 3.5.2. It can be concluded from the results in Table 3.6 that the two methods, when combined, reduce the run time effectively. Because they complement each other and produce good estimations of the optimal duals after the master problem is solved for the first iteration, the total number of iterations of the column generation process is reduced.

Table 3.6 Results of experiments on route initialization methods

Scenario	Initialization method	Run time (s)	#CG iterations
4	No initial routes	175	301
	Feasible planned routes	174	307
	Shortest path heuristic	213	337
	Feasible planned routes + Shortest path heuristic	145	262
5	No initial routes	317	427
	Feasible planned routes	294	384
	Shortest path heuristic	323	429
	Feasible planned routes + Shortest path heuristic	251	351

3.5.3.2 Continuous Delay vs. Discrete Delay

A large portion of the existing literature (Bisaillon et al., 2011; Bratu and Barnhart, 2006; Maher, 2016; Thengvall et al., 2003, 2001; Yan and Lin, 1997; Zhang et al., 2016) takes a discrete approach to modeling flight delay: flights are copied blindly and roughly with fixed intervals for all possible delay options. In contrast, in our approach, flight delay is a continuous variable and can be decided adaptively and accurately in the subproblems. In this section, we present the results of a comparison study between the two approaches modeling delay. To be fair, they are both solved in a column generation framework. As suggested by Maher (2016),

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

for the discrete delay approach, a flight is copied every 30 minutes until it reaches the maximum delay allowed. Table 3.7 and Figure 3.7 show the results of the comparison.

Table 3.7 A comparison of continuous delay and discrete delay approaches

Scenario	LP obj.		IP obj.		IP obj. gap between two approaches
	Discrete delay	Continuous delay	Discrete delay	Continuous delay	
1	1010.00	1010.00	1010	1010	0.00%
2	5513.00	4364.00	5513	4364	26.33%
3	6585.00	4781.00	6585	4781	37.74%
4	2156.67	2124.29	2160	2138	1.03%
5	230.00	218.00	230	218	5.50%

Figure 3.7 IP objective gap between discrete and continuous delay approaches

We can see from Table 3.7 and Figure 3.7 that, for most scenarios, there is a gap between the final IP objective values achieved by the discrete delay and continuous delay approaches. The discrete delay model provides an upper bound of the objective value for the continuous model because the discrete delay model carries out a more restrictive specification on delay

options. For some case, Scenario 1, for instance, the gap is zero because the optimal recovery solution does not require the delay of any flight, as shown in Table 3.4. However, for some other scenarios, Scenario 3 for example, the gap is as high as 37.74%, because many flights are subject to delay in the solution due to the substantial disruptive impact brought by ATC flow control at the hub airport. There could be large “error” caused by the discrete delay approach; on the contrary, our continuous delay approach can optimize flight delay precisely with zero modeling error.

3.5.3.3 Subproblem Parallelization

To further shorten the run time for potentially larger scenarios in future applications, we carried out computational experiments by more powerful computing machine that can use more threads to further parallelize subproblems and solve master problems more quickly. The program was switched to run on a workstation with two Intel Xeon 3.4 GHz CPUs. The maximum number of supported threads was 24. Scenario 5, the largest scenario, for instance, was solved by utilizing different numbers of threads. The relationship between the run time and number of used threads is recorded in Figure 3.8 and Table 3.8, where “MP time” and “SP time” stand for the times spent on master problems and subproblems, respectively. “IP time” is the time required for solving the final integer programming given all generated routes. Because the master problems were solved by CPLEX, their MP times are independent of the number of threads used to solve the subproblems. However, because of the increasing parallelization of subproblems, the run time spent on subproblems decreases rapidly, from 457 s to 145 s, amounting to a 67.76% drop. In theory, if the subproblems were completely parallelized, i.e., one thread for one subproblem, the run time would be $371/44 + 85 = 93.43$ s, which is about 1.5 minutes, to obtain the same very high-quality solution for the large-scale problem. It should be noted that the run time can be further reduced with a more professional computing environment for real industrial applications.

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

Figure 3.8 Run time with number of used threads run on a workstation (IP time is too small to be shown)

Table 3.8 Run time with number of used threads run on a workstation

No. of threads	MP time (s)	SP time (s)	IP time (s)	Run time (s)	Recovery cost
1	85	371	1	457	218
2	86	206	1	293	218
4	84	112	1	197	218
8	84	96	1	181	218
16	85	63	1	149	218
24	86	58	1	145	218

3.5.4 Computational Studies for Operational Insights

3.5.4.1 The Gain of Maintenance Flexibility

In this section, we investigate the gain brought by swapping planned maintenances. Two scenarios, Scenarios 6 and 7, are based on real airline operational data with distinct unplanned and planned maintenances, as illustrated in Table 3.9. The two scenarios are solved by the column generation framework with subproblems considering swappable planned maintenances, as described in Section 3.4.4.3. Within the 4-day planning horizon of the two scenarios, aircraft must be maintained at least every 60 flying hours, every 30 takeoffs, or every 70 hours, whichever happens first. The cost parameters are listed in Table 3.10. Cancelling and swapping planned maintenances does not incur any cost as long as the maximum flying hours, maximum cycles, and maximum interval constraints relating to the planned maintenances are satisfied. The computational settings are the same as those in Section 3.5.2. In addition, assuming that planned maintenances are not allowed to swap, we solve the two scenarios and compare the results with the ones that permit swapping planned maintenances, as shown in Table 3.11. Remarkably, for both scenarios, when planned maintenances are allowed to swap, the recovery cost is reduced considerably. Flight swaps and delays are reduced by utilizing maintenance swaps. For Scenario 6, the recovery cost drops by 19.78%, and for Scenario 7, it drops by 61.47%. Note that Scenario 7 has more planned maintenances than Scenario 6. It is recommended for airlines to consider maintenance swap as an option during aircraft recovery—a little flexibility would leverage big cost savings.

3.5.4.2 The Value of Airport Slot Quotas

Another operational insight which can be obtained from solving the ARP is regarding the value of airport slot quotas. As described in Section 3.2, when adverse weather occurs and ATC flow control is in place, airlines are allocated slot quotas specifying the maximum number of departures and arrivals the airline can have during flow control periods. In practice, the

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

Table 3.9 Scenarios with planned and unplanned maintenances

Scen- ario	No. of flights	No. of aircraft	No. of unplan. maint.	No. of plan. maint.	No. of airports	Hori- zon (h)	Disruption
6	586	44	24	33	37	85	Aircraft on ground for 48 hours
7	637	44	28	38	32	90	Aircraft on ground for 8 hours

Table 3.10 Cost parameters of Scenario 6 and Scenario 7

Cost parameters	Value
Swap per flight	10
Flight delay per minute	4
Cancellation per flight	500
Swap per planned maintenance	0
Cancellation per planned maintenance	0

Table 3.11 Comparison of recovery solutions given swappable and unswappable planned maintenances

KPIs	Scenario 6		Scenario 7	
	Plan. maint. unswappable	Plan. maint. swappable	Plan. maint. unswappable	Plan. maint. swappable
No. of flight cancellations	2	2	0	0
No. of flight swaps	124	110	41	29
No. of plan. maint. swaps	-	10	-	7
No. of plan. maint. cancellations	3	5	0	0
Total flight delay (min)	168	59	109	9
Recovery cost	2912	2336	846	326
Recovery cost reduction	19.78%		61.47%	

given quota can be traded with other airlines (ICAO, 2013). An airline may enlarge its limited maximum number of departures and arrivals to facilitate its recovery by buying slot quotas from others. Thus, the fundamental question becomes how much a slot quota is worth. For the airline that wants to purchase a slot quota, the answer depends on how much its recovery cost will be reduced if it is allowed one more departure or arrival during the slot period. The answer can be obtained by solving the ARP repeatedly with varying slot quotas. Take Scenario 3 for example, the ATC flow control takes place on a hub airport LEM (Lemmon Municipal Airport), from 4:00 pm to 5:00 pm, 7 May 2014. We keep increasing the maximum number of departures and arrivals from zero and record the corresponding recovery cost achieved after optimization. The computational settings are the same as those in Section 3.5.2, and the results are recorded in Table 3.12. The first row indicates the scenario in which the airline is granted no permission for taking off or landing flights during the ATC control period, which serves as a starting point for the computational experiment. It is observed from Table 3.12 that when the airline is given a higher departure and arrival quota, its recovery cost decreases monotonically. More remarkably, by and large, the marginal value of each additional quota amount continues to drop steadily. This implies that when the airline has already been allocated a sufficient departure and arrival quota, it is not economical to purchase any new slot quotas whose marginal value would be very small; the airline should consider other options to further reduce its recovery cost, for example, by giving out coupons to persuade passengers to wait longer than the prescribed maximum delay.

3.6 Summary

In this chapter, we considered the aircraft recovery problem with airport capacity constraints and maintenance flexibility. Disruptions have a considerable financial impact on the airline industry. When disruptions happen, airlines need to re-schedule flights and re-assign aircraft in real time with minimized recovery cost. In most published studies, airport capacity and flexible

Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility

Table 3.12 Marginal values of airport slot quotas

Airport slot quota	Recovery cost	Marginal value
0	4781	-
1	3778	1003
2	2774	1004
3	1968	806
4	1284	684
5	720	564
6	280	440
7	40	240

maintenances are not considered simultaneously via an optimization approach. To bridge this gap, we proposed a column generation framework to solve the problem. The framework consists of a master problem for selecting routes for aircraft and subproblems for generating routes. Airport capacity is explicitly considered in the master problem and swappable maintenances are incorporated in the subproblem. Instead of discrete delay models, which are widely adopted in much of the existing literature, flight delays are continuous and optimized accurately in the subproblems of this work. Compared to the discrete-delay method, the continuous-delay model improves the accuracy of the optimized recovery cost. In one tested scenario, the accuracy is improved by up to 37.74%. The computational study, which is based on real-world problems, shows that the master problem gives a very tight linear relaxation with a small, often zero, optimality gap. Large-scale problems can be solved within 5 minutes and the run time can be further shortened by parallel computing on more powerful hardware. In addition, from a managerial point of view, the computational experiments reveal that swapping planned maintenances may bring about a substantial reduction in recovery cost of up to about 60%, depending on problem instances. Furthermore, the decreasing marginal value of airport slot quotas is also highlighted for airline recovery operations through computational experiments.

Chapter 4

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

In this chapter, we investigate a coordinated multi-aircraft 4D trajectory planning problem considering buffer safety distance and fuel optimization in the context of 4D Trajectory Based Operations. Section 4.1 gives an introduction of the problem. Section 4.2 presents the contributions of this work. In Section 4.3, we present the discrete representation of airspace and 4D trajectories. Section 4.4 develops a model for the impact of buffer safety distance. Algorithms for coordinated multi-aircraft 4D trajectory planning are developed in Section 4.5, followed by a case study in Section 4.6. In Section 4.7, the results of comprehensive computational experiments to verify the proposed algorithm's computational performance are presented. Section 4.8 concludes the chapter.

4.1 Introduction

Each day, thousands of aircraft are flying in the sky. The Air Traffic Management system coordinates travelling aircraft to ensure safety and enhance efficiency. In the current ATM system, the airspace is divided into sectors. A team of air traffic controllers is assigned to be in charge of each sector. After take-off, an aircraft flies along airways traversing sectors one after another according to its flight plan under the direction of air traffic controllers responsible for each sector until its destination airport is approached.

With the development of the global economy, air traffic volume is undergoing rapid growth, particularly in emerging-market countries. The International Civil Aviation Organization estimates that air traffic in the Asia Pacific region will triple by 2030 (ICAO, 2012). The rapid growth of air traffic volume is placing significant challenges on the current clearance-based ATM system. To address such imminent challenges, both the USA and Europe have initiated research and development programs, namely, the Next Generation Air Transportation System (NextGen) (FAA, 2013) and the Single European Sky ATM Research (SESAR) (EUROCONTROL, 2015), respectively, for the future ATM system. Both programs envision 4D Trajectory Based Operations (4D TBO), in which 4D trajectories are calculated and followed using advanced navigation technologies. Aircraft will fly negotiated trajectories and air traffic control will move to trajectory management (FAA, 2013). Air traffic controllers will direct aircraft not only based on their current positions and speeds, but their future intended 4D trajectories as well.

Before an aircraft enters a sector, its airborne flight management system will calculate and submit its preferred estimated 4D trajectory for traversing the sector (FAA, 2016). Because the trajectories are planned individually and independently, they may cause conflicts or congestion, which leads to high systemwise cost. The ATM system, consisting of air traffic controllers, needs to coordinate the received 4D trajectories, carrying out necessary modifications or even

re-planning to ensure the safety of air traffic and improve its efficiency with the help of decision support tools.

In this chapter, we consider a multi-aircraft 4D trajectory planning problem in the en route phase and our aim is to provide a tool for air traffic controllers to change/re-plan trajectories for each individual aircraft traversing a sector that contains a 3D network of airways and waypoints. Given the required entry and exit waypoints as well as the expected entry time windows and exit times of aircraft, our proposed algorithm framework coordinates the aircrafts' preferred 4D trajectories and computes final 4D trajectories that ensure conflict-free operation and reduce overall fuel consumption and sector exit delay. To solve the problem, a negotiation-based coordinating process is proposed and modeled as a pure-strategy game among aircraft. The proposed negotiation does not need to be physically carried out among aircraft and air traffic controllers. It can be automatically realized as a part of the algorithm within the tool for facilitating timely decision making for air traffic controllers. The tool can efficiently compute 4D trajectories through an automatic negotiation process, which both guarantees safety and improves efficiency. Moreover, although the tool is developed for ATC, pilots and their airlines can also benefit from it because the tool takes aircraft's preferred trajectories into account and reduces both delay and fuel consumption, which creates an incentive for them to participate in the process.

Regarding air traffic safety, en route aircraft are required to keep a separation from each other throughout their flight. Two aircraft are defined to be in conflict if their relative distance is less than 5 nm (nautical miles) horizontally and (in China) 300 m vertically. The safety zone of a single aircraft is depicted in Figure 4.1 as a cylinder. If one aircraft is in the safety zone of another, the two aircraft are in conflict. The 5 nm horizontal separation is referred to as the safety distance, which is the minimum separation that two aircraft must keep if vertical separation is not in use. If separation is lost, the air traffic controllers in charge will face punishment. In real practice, air traffic controllers' priority is to ensure air traffic safety and

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

they intend to maintain a separation between aircraft several times greater than the required safety distance. The surplus portion of separation, called the buffer safety distance in this study, is used to accommodate potential emergencies, and as a critical tool for regulating air traffic controllers' workload (Averty et al., 2004; Djokic et al., 2010; Loft et al., 2007; Nagaoka and Brown, 2015). For example, Djokic et al. (2010) point out that, "The higher horizontal proximity, i.e., the closer the aircraft in the horizontal plane, the higher was controller workload." If aircraft are always kept far from each other, air traffic controllers will experience a relatively low workload. However, excessive buffer safety distance interferes with the efficiency of air traffic, resulting in unnecessary delay and sector capacity loss. Different air traffic controllers will maintain different amounts of buffer safety distance because of the variance within air traffic controllers' skills, experiences, and personalities. Therefore, a decision support tool is needed to allocate buffer safety distance and manage its tradeoff with air traffic efficiency. To address the problems relating to buffer safety distance, which is desirable in practical ATC operation, we propose an algorithm framework for 4D trajectory planning, that guarantees there will be no conflict and takes buffer safety distance into account.

Figure 4.1 Aircraft safety zone

4.2 Contributions of This Work

We summarize the contributions of this work as follows. First, we propose a coordination mechanism for online multi-aircraft 4D trajectory planning and model it as a pure-strategy game to effectively manage the tradeoff between operating cost and buffer safety distance. Second, a maximum improvement distributed algorithm (MIDA) is correspondingly designed and its convergence to a Nash equilibrium can be guaranteed by only solving single aircraft 4D trajectory planning problems when all the other aircraft's trajectories are given. Its efficiency and applicability to planning multi-aircraft 4D trajectories online are demonstrated by a case study with real air traffic peak-hour data, and MIDA's scalability is verified by comprehensive computational experiments for the predicted double and triple future traffic loads. Third, using a discrete airspace representation and discrete timeline, we model 4D trajectories as piecewise linear routes on the time-space network. An efficient dynamic programming algorithm is developed to jointly optimize fuel consumption, arrival delay, and buffer safety distance, with possible aircraft maneuvers including speed adjustments and flight level changes. Deconfliction is guaranteed not only at discrete time instances but also at any given time between them. Fourth, by conducting computational experiments using real traffic data and practical cost factors, we analyze the tradeoff between buffer safety distance and other operating costs resulting from fuel consumption and delay. Compared to the recorded trajectories in the data, the 4D trajectories planned by MIDA are estimated to reduce operating costs by about 16.7% with an even larger minimum separation.

4.3 Discrete Airspace Representation and Problem Formulation

In the current ATM system, en route civil aircraft fly along airways prescribed by the civil aviation authority from waypoint to waypoint. The airways are connected by waypoints and

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

form an airway network, as illustrated in Figure 4.2. On the other hand, vertically, each airway comprises several flight levels (Figure 4.3). Each flight level is unidirectional. Aircraft in odd flight levels (e.g., 7,500 m) fly from west to east, whereas aircraft in even flight levels (e.g., 7,800 m) fly from east to west.

Figure 4.2 Airway network of Xiamen airspace (Civil Aviation Authority of China, 2014)

Figure 4.3 Flight levels of one airway

4.3 Discrete Airspace Representation and Problem Formulation

In this study, we adopt a discrete approach to describe 4D trajectories due to its high flexibility and effectiveness of modeling. The planning horizon is discretized into time instances and airways are discretized by evenly interpolated virtual waypoints, as shown in Figure 4.4. Consider the time instance set $T = \{1, 2, \dots, n_t\}$, where $k \in T$ corresponds time $k \cdot \Delta T$ in which ΔT denotes the time step size. Moreover, the waypoint set is $D = \{1, 2, \dots, n_d\}$, which includes both physical and virtual waypoints. The distance between two neighboring waypoints in one flight level is Δd .

Figure 4.4 Discretized airways with waypoints and 4D legs

Two waypoints $i, j \in D$ are directly connected if aircraft are able to fly from waypoint i to j within one time step ΔT and comply with the minimum speed $v_{\min}$ and maximum speed $v_{\max}$ constraints specified by aircraft dynamics. We designate one such direct connection as one 4D leg $(i, j, k) \in D \times D \times T$, because it describes a 4D segment traveling from waypoint i to waypoint j within time period k to $k + 1$ as a part of an entire 4D trajectory. Note that waypoints are duplicated along parallel flight levels. Hence, flight level changes are modeled by vertical 4D legs, connecting two waypoints at two different flight levels. Furthermore, speed adjustments are made possible by selecting from among 4D legs of different physical lengths, i.e., the distance travelled in one time step, as illustrated in Figure 4.4.

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

Denote the set of 4D legs as E and consider a directed graph $G(D, E)$ whose nodes are waypoints D and arcs are 4D legs E . Furthermore, denote the set of aircraft as F . The multi-aircraft 4D trajectory planning problem is to a) decide the sector entry time δ_f and exit time α_f of every aircraft $f \in F$ and b) route every aircraft on the graph $G(D, E)$ from its given entry waypoint o_f to its exit waypoint d_f . Let $w_f(k)$ denote the waypoint visited by aircraft f at time instance k . Then, a 4D trajectory can be described by a sequence of waypoints with time $\{w_f(\delta_f), w_f(\delta_f + 1), \dots, w_f(\alpha_f)\}$, where $w_f(\delta_f) = o_f$ and $w_f(\alpha_f) = d_f$. Equivalently, a planned 4D trajectory of aircraft f can be viewed as a sequence of connected 4D legs $r_f = \{(i, j, k) | w_f(k) = i, w_f(k+1) = j, \text{ for } i, j \in D, k \in T\}$.

The discrete-time-discrete-space modeling approach is an approximation of the continuous airspace and continuous timeline. Continuous 4D trajectories are sampled at synchronized time instances, and the sampled physical positions are approximated by discrete waypoints. The advantages of the discrete method adopted in this study are several. First, conflict-free trajectories are completely guaranteed. As mentioned, 4D trajectories are modeled as sequences of parameterized 4D legs. Hence, conflict-free operation between any two 4D trajectories can be guaranteed if any two 4D legs from the two 4D trajectories are not in conflict. Because of the discrete-time-discrete-space modeling, the relative distance between any two 4D legs, not only at sampled instances but also at any time, can be calculated analytically, as detailed in Section 4.4. In Section 4.5, multi-aircraft 4D trajectory planning is first decomposed into single-aircraft 4D trajectory planning problems. For a single aircraft, the traveling cost of taking 4D legs that will lead to conflict with other aircraft is penalized by positive infinity. Hence, conflicting 4D legs are avoided. Second, a discrete method provides the flexibility to model complicated aircraft maneuvers. Because continuous trajectories are discretized, flight level changes and speed adjustments thus can be modeled as 4D legs as well; otherwise, complicated flight dynamics must be counted for every aircraft, which makes multi-aircraft trajectory planning almost intractable. Last but not least, a discrete-time-discrete-space model

converts continuous trajectory planning to a search over combinations of 4D legs, which opens a door to fast discrete algorithms such as dynamic programming.

4.4 Buffer Safety Distance and Intensity Function

In this study, we examine the effect of buffer safety distance on air traffic control when designing multi-aircraft 4D trajectories. It is widely acknowledged that the relative distances among aircraft play a critical factor in air traffic controllers' workload (Averty et al., 2004; Loft et al., 2007). ATC real-time simulations also corroborate that aircraft horizontal proximity strongly correlates with air traffic controllers' workload (Djokic et al., 2010). Air traffic controllers' perception of the difficulty of an air traffic situation depends on the proximity of aircraft in the sector airspace (Nagaoka and Brown, 2015). If separations greater than the safety distance are always maintained (i.e., buffer safety distances always exist), air traffic controllers experience less intense traffic situations that need to be managed.

According to the Weber-Fechner law in psychology, the resolution of perception diminishes for stimuli of greater magnitude (Johnson et al., 2002). In the ATC context, the effect of buffer safety distance decreases as aircraft are separated further. Bearing this in mind, we developed an intensity scale function of the relative distance between two aircraft to capture the non-linear effect of buffer safety distance. The value of the intensity scale function is then integrated over time to reflect its impact over the planning horizon. The calculated intensity is included in the objective function of the algorithm framework, as demonstrated in Section 4.5. Consequently, the algorithm penalizes a small separation between aircraft more severely than a large one.

More specifically, consider two aircraft, aircraft 1 and aircraft 2, flying their 4D legs respectively within one flight level, as shown in Figure 4.5. Aircraft 1 is flying from waypoint i to j and aircraft 2 is flying from waypoint p to q from time instance k to $k + 1$. Denote the period start time as $t_s = k\Delta T$ and the finish time as $t_e = (k + 1)\Delta T$. The coordinates of aircraft 1 at waypoint i are P^1 , and aircraft 1's speed is V^1 ; the coordinates of aircraft 2 at waypoint p are P^1 , and aircraft

Figure 4.5 Two aircraft flying two 4D legs

2's speed is V^2 . Then, the relative start position of the two aircraft is $P = P^1 - P^2$. Similarly, the relative speed of the two aircraft is $V = V^1 - V^2$. Hence, the relative distance between the two aircraft at any time $t \in [t_s, t_e]$ is $d(t) = |P + (t - t_s)V| = \sqrt{(P + (t - t_s)V)^T (P + (t - t_s)V)} = \sqrt{V^T V (t - t_s)^2 + 2P^T V (t - t_s) + P^T P}$. The minimum value $d_{\min}$ of $d(t)$ can be solved analytically as follows. Denote the inside quadratic function as $f(t) \triangleq V^T V (t - t_s)^2 + 2P^T V (t - t_s) + P^T P$. Take the derivative of $f(t)$ and set it to 0 as follows: $\frac{df(t)}{dt} = 2V^T V (t - t_s) + 2P^T V = 0 \Rightarrow t = -\frac{P^T V}{V^T V} + t_s$. Let $t^* \equiv \arg \min_{t \in [t_s, t_e]} d(t)$. There are three possible cases for computing t^* . If $t_s < -\frac{P^T V}{V^T V} + t_s < t_e$, then $t^* = -\frac{P^T V}{V^T V} + t_s$; if $-\frac{P^T V}{V^T V} + t_s \leq t_s$, then $d(t)$ is monotonically increasing on $[t_s, t_e]$, so $t^* = t_s$; otherwise, $d(t)$ is monotonically decreasing on $[t_s, t_e]$, so $t^* = t_e$. Once t^* is derived, we can simply have $d_{\min}$ as $d(t^*)$.

Denote s as the safety distance. If $d_{\min} < s$, the two 4D legs (i, j, k) and (p, q, k) are in conflict. Denote the intensity induced by the aircraft flying the two 4D legs (i, j, k) and (p, q, k) as ϕ_{ijpq} . Define $\phi_{ijpq} = +\infty$ if $d_{\min} < s$, so the traveling cost on conflicting 4D legs is penalized by $+\infty$ in the 4D trajectory planning algorithm introduced in Section 4.5, and conflict-free trajectories can be guaranteed; otherwise, define $\phi_{ijpq} = \int_{t_s}^{t_e} \varphi(d(t)) dt$, where $\varphi(d)$ is the intensity scale function, which is designed as an exponential function depicted in Figure 4.6. The intuition behind taking the integral of the intensity scale function $\varphi(d)$ over time

4.4 Buffer Safety Distance and Intensity Function

Figure 4.6 Intensity scale function (s denotes one safety distance)

comes from the surveillance task that air traffic controllers perform when they are on duty. In real practice, air traffic controllers must keep monitoring all the aircraft in the sector all the time. Intensity is accumulated over time. The intensity scale function is used to describe the non-linear effect of buffer safety distance on the perceived intensity of the traffic situation faced by air traffic controllers. It decreases rapidly as the separation between two aircraft increases. The convexity of the function means that, when two aircraft are close, the increase of buffer safety distance causes significant decrease in intensity; in contrast, when the separation is large (e.g., three times the safety distance), increasing the buffer safety distance has little effect.

Some remarks: (a) The choice of the intensity scale function is not unique. Here, it takes an exponential form, as suggested by the Weber-Fechner law (Johnson et al., 2002). It must be a decreasing function since in real life the intensity scale decreases as the buffer safety distance increases. Other possible choices are, for example, step-wise decreasing functions, which mean in a certain range, the increase of the buffer safety distance does not change the intensity scale, but beyond the range, the intensity decreases in a step manner. (b) For 4D legs on two different

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

flight levels, their intensity $\varphi(d)$ is zero, because aircraft flying at different flight levels are clearly deconflicted. (c) Vertical 4D legs are projected to their neighboring flight levels for the intensity computation. When an aircraft is climbing or descending via a vertical 4D leg, the other 4D legs conflicting with the vertical 4D leg will be blocked due to the huge intensity induced.

4.5 Coordinated Multi-aircraft 4D Trajectory Planning

4.5.1 4D Trajectory Planning as a Pure-Strategy Game

Nowadays Collaborative Decision Making has been widely adopted in ATC practices ([FAA, 2014](#)). The main idea of CDM is to share decision-making responsibility between ATC and airspace users, so airspace users have more control over the decision. Airspace users can carry out negotiations with each other for the airspace resource and make economic tradeoffs according to their own interests which the air traffic controllers are not fully aware of. Under CDM, game theory becomes a suitable tool in modeling the 4D trajectory planning problem because the interactions among individual airspace users can be explicitly formulated and analyzed.

In future 4D Trajectory Based Operations, before entering a sector, aircraft submit their preferred 4D trajectories to the ATM system for coordination ([FAA, 2016](#)). From the air traffic controllers' point of view, the coordination simulates a negotiation process among aircraft entering and flying within the sector. Each aircraft, as a party in the negotiation, aims to minimize its own operating cost using its preferred 4D trajectory; meanwhile, after the 4D trajectories proposed by the other aircraft are revealed, a compromise should be made to modify the submitted 4D trajectory to resolve conflicts and alleviate congestion. Such compromise reflects the spirit of coordination: selfish individuals making compromises to optimize the entire system's performance. After every aircraft re-optimizes and re-submits its own 4D

4.5 Coordinated Multi-aircraft 4D Trajectory Planning

trajectory given those of the others, the negotiation process repeats. The coordinated 4D trajectory planning process forms a pure-strategy routing game in which players are aircraft, and the strategies are the all possible candidate 4D trajectories that each aircraft can choose from. The resolution of this routing game is an equilibrium joint routing decision, at which no aircraft is able to further improve its own 4D trajectory unilaterally.

Mathematically, suppose the aircraft set is $F = \{1, 2, \dots, n_f\}$ and each aircraft f 's all possible 4D trajectories are in its set R_f , $|R_f| = m_f$. Denote the 4D trajectory chosen by aircraft f as $r_f \in R_f$. Given the 4D trajectories chosen by the other aircraft, i.e., the 4D trajectory combination $(r_1, r_2, \dots, r_{n_f})$, the cost to aircraft f is

$$\pi_f(r_1, r_2, \dots, r_{n_f}) = c_{r_f} + \phi_{r_f r_1} + \phi_{r_f r_2} + \dots + \phi_{r_f r_{n_f}} = c_{r_f} + \sum_{g \in F} \phi_{r_f r_g},$$

where c_f is the operating cost (fuel plus delay) of aircraft f flying its candidate 4D trajectory r_f and $\phi_{r_f r_g}$ is the intensity between aircraft f 's 4D trajectory r_f and aircraft g 's 4D trajectory r_g . The corresponding c_{r_f} and $\phi_{r_f r_g}$ can be computed according to the 4D legs comprising r_f and r_g .

A Nash equilibrium is a 4D trajectory combination $(r_1^*, r_2^*, \dots, r_{n_f}^*)$ which satisfies

$$\pi_f(r_1^*, r_2^*, \dots, r_{n_f}^*) \leq \pi_f(r_1^*, \dots, r_{f-1}^*, r_f, r_{f+1}^*, \dots, r_{n_f}^*), \quad \forall f \in F, \forall r_f \in R_f.$$

It can be proved that a Nash equilibrium always exists for the formulated pure-strategy routing game of 4D trajectory planning.

Theorem 4.1. *The formulated pure-strategy routing game has at least one Nash equilibrium.*

Proof. Consider an integer quadratic programming (IQP) in which decision variables are as follows.

$$x_{r_f} = \begin{cases} 1 & \text{if aircraft } f \text{ chooses to fly 4D trajectory } r_f \\ 0 & \text{otherwise} \end{cases}$$

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

The objective function and constraints are:

$$\min \sum_{f \in F} \sum_{r_f \in R_f} c_{r_f} x_{r_f} + \sum_{f \in F} \sum_{r_f \in R_f} \sum_{g \in F} \sum_{r_g \in R_g} \phi_{r_f r_g} x_{r_f} x_{r_g} \quad (4.1)$$

$$\text{s.t.} \quad \sum_{r_f \in R_f} x_{r_f} = 1, \forall f \in F \quad (4.2)$$

$$x_{r_f} \in \{0, 1\}, \forall f \in F, r_f \in R_f \quad (4.3)$$

The objective function of Equation 4.1 minimizes the overall cost of the chosen 4D trajectories for the group of aircraft under planning. The constraints of Equation 4.2 mean that every aircraft chooses one and only one 4D trajectory, and the constraints of Equation 4.3 consist of the binary constraints for the decision variables. Let $\{x_{r_f}^*\}$ be the optimal solutions of the IQP, and the corresponding selected 4D trajectories are $\{r_f^*\}$, i.e., the selected 4D trajectory combination is $(r_1^*, r_2^*, \dots, r_{n_f}^*)$, which should be a Nash equilibrium.

Assume, on the contrary, that $(r_1^*, r_2^*, \dots, r_{n_f}^*)$ is not a Nash equilibrium. Then, for some aircraft $h \in F$, it has a 4D trajectory $\hat{r}_h \in R_h$ that satisfies

$$\pi_h(r_1^*, \dots, r_h^*, \dots, r_{n_f}^*) > \pi_h(r_1^*, \dots, \hat{r}_h, \dots, r_{n_f}^*) \iff c_{r_h^*} + \sum_{g \in F} \phi_{r_h^* r_g^*} > c_{\hat{r}_h} + \sum_{g \in F} \phi_{\hat{r}_h r_g^*},$$

which implies that the optimal objective function value of the formulated IQP holds such that

$$\begin{aligned} J^* &= \sum_{f \in F} \sum_{r_f \in R_f} c_{r_f} x_{r_f}^* + \sum_{f \in F} \sum_{r_f \in R_f} \sum_{g \in F} \sum_{r_g \in R_g} \phi_{r_f r_g} x_{r_f}^* x_{r_g}^* \\ &= \sum_{f \in F} c_{r_f^*} + \sum_{f \in F} \sum_{g \in F} \phi_{r_f^* r_g^*} \\ &= \sum_{\substack{f \in F \\ f \neq h}} c_{r_f^*} + \sum_{\substack{f \in F \\ f \neq h}} \sum_{g \in F} \phi_{r_f^* r_g^*} + c_{r_h^*} + \sum_{g \in F} \phi_{r_h^* r_g^*} \end{aligned}$$

$$> \sum_{\substack{f \in F \\ f \neq h}} c_{r_f^*} + \sum_{\substack{f \in F \\ f \neq h}} \sum_{g \in F} \phi_{r_f^* r_g^*} + c_{\hat{r}_h} + \sum_{g \in F} \phi_{\hat{r}_h r_g^*} \triangleq \hat{J}.$$

Here, $\hat{J}$ is achieved by $\{\hat{x}_{r_f}\}$ corresponding the 4D trajectory choice of aircraft h selecting $\hat{r}_h$ rather than r_h^* . Solution $\{\hat{x}_{r_f}\}$ is a better solution than $\{x_{r_f}^*\}$, which contradicts the optimality of $\{x_{r_f}^*\}$. Hence, 4D trajectory combination $(r_1^*, r_2^*, \dots, r_{n_f}^*)$ is a Nash equilibrium. $\square$

It should be noted that the IQP formulated above is only for the proof of Theorem 4.1, i.e., the existence of at least one Nash equilibrium. Although, in theory, a Nash equilibrium can be obtained from enumerating all possible 4D trajectories and solving the IQP, in practice, the IQP can hardly be used for practical online multi-aircraft 4D trajectory planning because the number of candidate 4D trajectories is tremendously large; the IQP is computationally intractable for real-sized problems. Hence, we next propose a practical distributed algorithm to find an equilibrium solution for 4D trajectory planning, and its efficacy is validated in the case study presented in Section 4.6.

4.5.2 Maximum Improvement Distributed Algorithm

The Maximum Improvement Distributed Algorithm (MIDA) serves as a protocol for the negotiation process among aircraft. It specifies at the end of each iteration of negotiation which aircraft takes the advantage to update its 4D trajectory and the others make compromises. MIDA mimics the procedure of an auction. During each iteration of the algorithm, every aircraft bids to change its current 4D trajectory. The aircraft that not only improves its own cost but at the same time brings the most reduction to the systemwise overall cost is entitled to change. This process continues until equilibrium is reached when no aircraft has the incentive to adjust its 4D trajectory unilaterally. Notably, instead of searching among all the possible 4D trajectories of each aircraft, as in the IQP in the proof of Theorem 4.1, MIDA generates one

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

new candidate 4D trajectory for each aircraft in each iteration as a subproblem. Subproblems are distributed among aircraft and can be solved efficiently, which makes MIDA fast at finding an equilibrium.

The framework of MIDA is summarized in Algorithm 4.1. MIDA begins with a set of 4D trajectories which can be initialized by individual single-aircraft 4D trajectory planning regardless of other aircraft. Then, given all other aircraft's 4D trajectories which are fixed, each aircraft optimizes its own 4D trajectory as a single-aircraft 4D trajectory optimization problem. The aircraft with the highest reduction in the cost function π_f is chosen to update its 4D trajectory. If no aircraft can reduce its cost function π_f any further, the algorithm stops. The resulting 4D trajectory combination is a Nash equilibrium, as shown in Theorem 4.2.

Algorithm 4.1 Framework of the maximum improvement distributed algorithm

```

Initialize 4D trajectory  $r_f^1$  for each aircraft  $f \in F$ 
for iteration  $n = 1, 2, \dots$  do
    for aircraft  $f = 1, 2, \dots, n_f$  do
        Generate  $r_f^*$  which minimizes  $\pi_f(r_1^n, \dots, r_f, \dots, r_{n_f}^n)$ 
        Save  $\Delta\pi_f = \pi_f(r_1^n, \dots, r_f^*, \dots, r_{n_f}^n) - \pi_f(r_1^n, \dots, r_f^n, \dots, r_{n_f}^n)$ 
    end for
    Find  $g = \arg \min_f \Delta\pi_f$ 
    if  $\Delta\pi_g < 0$  then
        Update  $r_g^{n+1} = r_g^*$ , and  $r_f^{n+1} = r_f^n$  for  $\forall f \neq g$ 
    else
        Exit
    end if
end for

```

One remarkable advantage of MIDA is its scalability with the number of aircraft involved in 4D trajectory planning. Due to its distributed algorithmic feature, in each iteration, the centralized multi-aircraft trajectory planning is decomposed into n_f subproblems, and each subproblem is a single-aircraft 4D trajectory optimization problem in which r_f^* is generated. Moreover, the single-aircraft 4D trajectory planning problem can be equivalently formulated and solved as a shortest path problem, as detailed in Section 4.5.3 below. When more aircraft

4.5 Coordinated Multi-aircraft 4D Trajectory Planning

are involved, the per-iteration computation load of each aircraft does not change much. The case study in Section 4.6 confirms its efficiency at solving real-sized multi-aircraft 4D trajectory planning problems.

For the completeness of algorithm design, next, we prove the convergence of MIDA regardless of how the aircraft 4D trajectories are initialized.

Theorem 4.2. *MIDA converges to a Nash equilibrium in a finite number of iterations.*

Proof. Consider iteration n , at which the 4D trajectory combination is $\Gamma^n = (r_1^n, r_2^n, \dots, r_{n_f}^n)$, and the corresponding systemwise cost is

$$J(\Gamma^n) = \sum_{f \in F} c_{r_f^n} + \sum_{f \in F} \sum_{g \in F} \phi_{r_f^n r_g^n}$$

After the iteration, suppose aircraft h 's 4D trajectory is updated from r_h^n to r_h^{n+1}

$$J(\Gamma^{n+1}) = \sum_{\substack{f \in F \\ f \neq h}} c_{r_f^n} + \sum_{\substack{f \in F \\ f \neq h}} \sum_{g \in F} \phi_{r_f^n r_g^n} + c_{r_h^{n+1}} + \sum_{g \in F} \phi_{r_h^{n+1} r_g^n}$$

The overall improvement is

$$\begin{aligned} J(\Gamma^{n+1}) - J(\Gamma^n) &= (c_h^{n+1} + \sum_{g \in F} \phi_{r_h^{n+1} r_g^n}) - (c_h^n + \sum_{g \in F} \phi_{r_h^n r_g^n}) \\ &= \pi_h(r_1^n, \dots, r_h^{n+1}, \dots, r_{n_f}^n) - \pi_h(r_1^n, \dots, r_h^n, \dots, r_{n_f}^n) < 0 \end{aligned}$$

Hence, after each iteration, 4D trajectory combination Γ^n is updated and the systemwise cost $J(\Gamma^n)$ strictly decreases. Combination Γ^n cannot revert to any $\Gamma^{\hat{n}}$, $\hat{n} < n$, which has already been iterated. Because the number of possible 4D trajectories r_f is finite, the number of possible 4D trajectory combinations Γ is finite. In the worst case, every possible Γ is iterated and MIDA stops. MIDA must stop within a finite number of iterations. When it stops after iteration n ,

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

MIDA's termination condition ensures that

$$\begin{aligned} \Delta\pi_f &= \pi_f(r_1^n, \dots, r_f, \dots, r_{n_f}^n) - \pi_f(r_1^n, \dots, r_f^n, \dots, r_{n_f}^n) \geq 0, \quad \forall r_f \in \mathbf{R}_f, f \in \mathbf{F}, \\ \text{i.e.,} \quad \pi_f(r_1^n, \dots, r_f^n, \dots, r_{n_f}^n) &\leq \pi_f(r_1^n, \dots, r_f, \dots, r_{n_f}^n), \quad \forall r_f \in \mathbf{R}_f, f \in \mathbf{F}. \end{aligned}$$

Therefore, it reaches a Nash equilibrium. □

We have shown the existence of a Nash equilibrium and MIDA's computational convergence. Moreover, it will be helpful to note that the Nash equilibrium is not unique in this game. As shown in Theorem 4.1, the global optimal solution of the IQP constructed in the proof must be a Nash equilibrium. However, a Nash equilibrium is not necessarily the global optimal solution because of the “non-convexity” of the IQP objective function. The corresponding objective function 4.1 can be equivalently reformulated as $\mathbf{c}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x}$, where $\mathbf{Q}$ is a symmetric matrix and represents the intensity between 4D trajectories. For the 4D trajectory planning problem in the IQP form, all the diagonal elements of $\mathbf{Q}$ are zero because the intensity is only defined for two different trajectories. Furthermore, all the other elements in $\mathbf{Q}$ must be nonnegative because the intensity is nonnegative. Based on the two conditions above, matrix $\mathbf{Q}$ must not be positive semi-definite, which is the necessary and sufficient condition for the convexity of objective function 4.1. Therefore, there exist multiple local optimal points which may also be Nash equilibriums. Although the Nash equilibrium found by MIDA cannot guarantee to be the global optimal solution, it can still largely improve the system performance, as shown in the case study in Section 4.6. On the other hand, if run-time restrictions are relaxed, the optimality of MIDA can be enhanced by several ways. For example, multiple randomized initialization can be adopted for approaching the global optimal. Alternatively, MIDA is possibly combined with other meta-heuristics such as genetic algorithms or simulated annealing. The global optimal

solution will be reached with probability one by the combined algorithm because the search space is finite.

4.5.3 Single-Aircraft 4D Trajectory Optimization

Single-aircraft 4D trajectory optimization is a fundamental step of MIDA. Its efficiency determines the algorithm's overall computational performance. In this section, we demonstrate that the single-aircraft 4D trajectory optimization problem can be formulated as a shortest path problem and hence solved in polynomial time. The problem not only determines the 3D trajectory that every aircraft passes, but also the time at which each waypoint is visited.

The problem is stated as follows. Given the other aircraft's 4D trajectories r_g for any $g \neq f$, aircraft f decides its sector entry time δ_f and exit time α_f , and generates a 4D trajectory r_f , which is a sequence of connecting arcs (4D legs) on $G(D, E)$, with the objective of minimizing the sum of the costs of fuel, delay, and intensity. Aircraft f 's entry waypoint is o_f and exit waypoint is d_f . It must enter the sector within time window $[k_f^s, k_f^e]$, and the expected exit time is k_f^a . The planned 4D trajectory must ensure there is no conflict with all other aircraft's given trajectories not only at time instances $k \in T$ but also any time between them.

To take time dimension explicitly into account, $G(D, E)$ is duplicated over the time instances to construct a time-space graph, which is sketched in Figure 4.7. Figure 4.7 takes one duplicated airway as an example. Every layer corresponds to one time instance. Nodes are waypoints duplicated on the time instances and arcs across layers are 4D legs. Two dummy nodes (source s^+ and sink s^-) are added to facilitate the decision of the sector entry time and exit time. Node s^+ is connected to every duplicated entry point o_f within $[k_f^s, k_f^e]$, and s^- is connected to every duplicated exit waypoint d_f .

Denote c_{ijk} as the cost of travelling on 4D leg (i, j, k) . Here, $c_{ijk} = \omega_1 c_{ij} + \omega_2 \phi_{ijk}$, where c_{ij} is the fuel consumption cost for an aircraft travelling from waypoint i to waypoint j in one time step ΔT ; ϕ_{ijk} is the intensity cost induced by the proximity between aircraft f and the

Figure 4.7 A sketch of time-space graph

other aircraft if aircraft f flies from waypoint i to j during time step k to $k + 1$; and ω_1 and ω_2 are weights that can be tuned according to application needs. For the arcs connecting d_f of time instance k and the sink s^- , $c_{d_f s^- k}$ is defined as the delay cost, i.e., $c_{d_f s^- k} = \omega_3(k - k_f^a)$ where $k - k_f^a$ is exit time delay and ω_3 is weight. For arcs connecting source s^+ and o_f of time instance k , cost $c_{s^+ o_f 0} = 0$.

After travelling cost c_{ijk} is associated with each 4D leg, the single-aircraft 4D trajectory optimization problem reduces to finding the shortest path between s^+ and s^- . The problem can be efficiently solved by the dynamic programming algorithm presented in Algorithm 4.2. Its computational complexity is linear to the total number of 4D legs.

Algorithm 4.2 A dynamic programming for single-aircraft 4D trajectory planning

```

Initialize node value  $\Psi(s^+) = 0, \Psi(s^-) = +\infty, \Psi(i, k) = +\infty, \forall i \in D, k \in T$ 
for each 4D leg  $(i, j, k)$  do
    if  $\Psi(j, k + 1) > \Psi(i, k) + c_{ijk}$  then
        Update  $\Psi(j, k + 1) = \Psi(i, k) + c_{ijk}$ 
        Mark  $i$  as the precedent waypoint of  $j$ 
        if  $j = s^-$  then
            Update sector exit time  $\alpha_f = k$ 
        end if
    end if
end for

```

4.6 Case Study Based on Real Traffic Data

In this section, we apply MIDA to solving 4D trajectories based on real air traffic data. The sector under study is ZSAMAR01 in Xiamen area, one of the busiest ATC sectors in the southeastern part of China. A horizontal view of the airway network of the sector is shown in Figure 4.8. The air traffic data contain information of aircraft passing through the sector from 2:00 pm to 3:00 pm on 7 August 2008, which is the peak hour of the day: every 4 s, all in-sector aircraft's 3D locations (3D coordinates) are sampled and stored in the database; therefore, the real 4D trajectories of all aircraft are recorded. In total, 31 passing aircraft were

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

included in the planning horizon. The aircraft fly at three flight levels (7,800, 7,500, and 7,200 m) in the upper space of the sector. The sector serves as a major transitional airspace between airport terminal area and higher en-route airspace. For example, for aircraft taking off from Xiamen airport and flying north (e.g., to Beijing), they follow airways P84-DO and DO-P48 before climbing to higher sectors in the middle way of DO-P48 under the direction of air traffic controllers, as illustrated in Figure 4.9. In the model, vertical 4D legs connecting flight levels are added to facilitate aircraft flight level changes, as described in Section 4.3.

Figure 4.8 Xiamen ZSAMAR01 sector

The parameter settings for the case study are as follows: time step $\Delta T = 1$ minute, the distance between neighboring waypoints $\Delta d = 1$ nm. Aircraft can change their speeds among 360, 420, and 480 knot. Sector entry points o_f and sector exit points d_f were extracted from

Figure 4.9 Flight levels and a sample aircraft trajectory

traffic data. Entry time windows $[k_f^s, k_f^e]$ and expected times of arrival k_f^a were set according to the data flight plans. Fuel consumption c_{ij} , which is the fuel consumed flying from waypoint i to waypoint j in one time step ΔT , was derived from the Base of Aircraft Data (BADA) developed by [EUROCONTROL \(2016\)](#). The fuel consumption curves of A320, which is one of the most commonly used civil aircraft types, is shown in Figure 4.10.

Weights, ω_1 for fuel consumption and ω_3 for sector exit delay, can be set according to financial analysis. Following [Soler et al. \(2014\)](#), the unit fuel cost was estimated to be 1.30 USD/kg, and the CO₂ emission cost for each kilogram of fuel is 0.11 USD/kg. Hence, ω_1 equals 1.41 USD/kg, the sum of the two costs. For the cost of delay, assuming 21.35 USD/hour as the average value of a passenger's time ([Soler et al., 2014](#)) and a typical 150-seat A320 aircraft with 80% occupancy, the cost of 1 minute of delay of one aircraft ω_3 amounts to 42.7 USD/minute. Weight ω_2 for intensity is difficult to quantify. Roughly speaking, it should be linked to air traffic controllers' scaled salary, taking workload into consideration ([Sherali et al.,](#)

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

Figure 4.10 Fuel consumption curves of A320

2006), or it can be considered as the safety cost of different intensity scales. In this study, our aim is to develop a 4D trajectory planning tool to coordinate aircraft traversing a sector for air traffic controllers. As the tool's users, air traffic controllers can set the intensity weight ω_2 freely as a parameter of the tool; ω_2 reflects the heterogeneity within air traffic controllers: conservative air traffic controllers tend to specify larger ω_2 to encourage higher buffer safety distance. The effect of ω_2 on intensity and aircraft separation is studied by numerical tests below.

Given the parameters discussed above, MIDA was run for scenarios corresponding to different settings of intensity weight ω_2 . Each aircraft's 4D trajectory was initialized by single-aircraft 4D trajectory optimization regardless of the other aircraft. MIDA was programmed in C++ running on a desktop computer with a 3.9 GHz Intel i3 CPU and 32 GB RAM in Windows 10 OS. The computational results are summarized in Table 4.1 and plotted in Figure 4.11. All scenarios were solved in about 1 s. In Table 4.1, the "Monetary cost" column equals

4.6 Case Study Based on Real Traffic Data

the fuel and delay costs converted to USD by cost factors ω_1 and ω_3 . “Min separ.” stands for the minimum horizontal relative distance that occurs between any pair of aircraft not separated vertically through the planned 4D trajectories. Furthermore, the total time during which aircraft separation was less than three times the safety distance was computed and is presented in “Dur. in 3*s.” The “Delay” column sums up all aircraft’s sector exit delays.

Figure 4.11 Impact of intensity weight on 4D trajectories intensity and minimum separation

Several observations can be made from Table 4.1. First, with increasing cost weights for intensity, the separations among planned 4D trajectories continue to increase in a stepwise manner (Figure 4.11). When the cost weight is low, in Scenario 1 for example, the minimum separation is 7 nm, with only a 2 nm buffer safety distance. In contrast, after the cost weight is increased to 70, the minimum separation distance increases to 11 nm, which is more than twice the safety distance (5 nm). Second, as indicated by the “Dur. in 3*s” column, for all planned 4D trajectories, the duration for which some aircrafts’ in-effect horizontal separation is less

Table 4.1 Results of multi-aircraft 4D trajectory planning under different intensity weights

Scenario	Intensity		Cost		Separation		Run time (s)	
	weight	Fuel (kg)	Delay (min)	Monetary cost (\$)	Intensity	Min separ. (mm)		
1	10	10637.90	0	14999.44	1.01	7.00	1.23	1.00
2	20	10640.66	0	15003.33	0.79	8.00	1.16	1.06
3	30	10650.18	0	15016.75	0.34	9.00	0.79	0.99
4	40	10652.93	0	15020.64	0.23	10.00	0.66	0.99
5	50	10652.93	0	15020.64	0.23	10.00	0.66	1.00
6	60	10652.93	0	15020.64	0.23	10.00	0.66	0.98
7	70	10655.69	0	15024.53	0.17	11.00	0.53	0.98
8	80	10655.69	0	15024.53	0.17	11.00	0.53	1.00
9	90	10655.69	0	15024.53	0.17	11.00	0.53	0.98
10	100	10655.69	0	15024.53	0.17	11.00	0.53	0.98
data	-	10274.12	83	18035.54	0.36	8.36	1.03	-

4.6 Case Study Based on Real Traffic Data

than three times the safety distance is very small, less than 2 minutes, compared to the whole length of the planning horizon of one hour. In other words, most of the time, aircraft keep a buffer safety distance that is three times more than required. Finally, because of the relatively large weight for the delay, all aircraft's exit delay costs are minimized to zero and the delay cost does not change with the increasing intensity weight, whereas fuel consumption cost rises mildly. It can be concluded from the "Fuel" and "Min separ." columns of Scenarios 2 and 3 that every additional nautical mile of buffer safety distance in minimum separation costs at most $10,650.18 - 10,640.66 = 9.52$ kg fuel, which is worth USD 13.42.

Compared to the real trajectories recorded in the data (the metrics computed in the last row of Table 4.1), which are mainly based on the first-come-first-served ATC practice, all planned 4D trajectories by MIDA achieve less monetary cost and most maintain a larger minimum separation and lower intensity. In fact, the intensity and minimum separation of the real trajectories recorded in the data are mainly caused by an encounter of two aircraft, CAA1383 and HXA2656, near the busy crossover waypoint DO. In the beginning, the two aircraft fly towards each other on neighboring flight levels 7,500 and 7,200 m, so deconfliction is maintained vertically. After they pass by each other, as shown in Figure 4.12(a), when HXA2656 begins to climb, vertical separation is lost, and deconfliction is only maintained by horizontal separation, which is 8.4 nm with just a 3.4 nm buffer safety distance. In contrast, according to the 4D trajectories planned by MIDA, Scenario 7 for example, HXA2656's sector entry time is delayed by 1 minute, and therefore, when it climbs, CAA1383 is further away; the horizontal separation increases to 12.7 nm, more than twice the safety distance, as illustrated in Figure 4.12(b).

The case study shows that, although most of the time in real practice aircraft are separated by abundant buffer safety distance, due to converging airways or improper timing, air traffic controllers still sometimes face intense air traffic situations where more buffer safety distance would be preferred. The algorithm framework developed in this work is able to serve as a

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

Figure 4.12 An encounter of two aircraft (a) recorded in data (b) planned by MIDA

decision making tool for air traffic controllers or an ATM system to plan 4D trajectories while considering buffer safety distance in a realistic application context. Compared to the real trajectories recorded in the air traffic data, the planned 4D trajectories are estimated to reduce monetary cost by 16.7%, with less intense air traffic situations and larger minimum separation. On the other hand, from a mathematical point of view, although MIDA does not guarantee a global optimal solution as stated in Section 4.5.2, we may derive a lower bound for the objective value of the optimal solution as follows. First, each aircraft plans its own optimal 4D trajectory regardless of others as the single-aircraft 4D trajectory planning problem described in Section 4.5.3. Then, sum up all aircraft's fuel and delay costs and exclude the intensity costs. Thus, the lower bound is obtained and we can have an estimated gap between the global optimal solution and the solution by MIDA. The true optimality loss must be less than the estimated gap. For the case study scenarios, the calculated lower bound is \$14798.56, the estimated gaps are listed in Table 4.2.

4.7 Computational Experiments for Future Traffic Demand

Table 4.2 Estimated gaps to the global optimal solutions

Scenario	Weight	Monetary cost (\$)	Intensity	Overall cost	Estimated gap
1	10	14999.44	1.01	15009.49	1.43%
2	20	15003.33	0.79	15019.23	1.49%
3	30	15016.75	0.34	15026.82	1.54%
4	40	15020.64	0.23	15029.89	1.56%
5	50	15020.64	0.23	15032.21	1.58%
6	60	15020.64	0.23	15034.52	1.59%
7	70	15024.53	0.17	15036.23	1.61%
8	80	15024.53	0.17	15037.91	1.62%
9	90	15024.53	0.17	15039.58	1.63%
10	100	15024.53	0.17	15041.25	1.64%

4.7 Computational Experiments for Future Traffic Demand

In this section, we present the results of computational experiments carried out to investigate the computing speed of MIDA given the predicted double or triple traffic volumes in the near future (ICAO, 2012). Because MIDA is designed as an online tool for planning 4D trajectories of multiple aircraft entering a sector, its scalability in terms of the increasing number of aircraft is vital for practical applications. In addition, MIDA must maintain proper separation among computed 4D trajectories, even in the more complicated traffic situations when more aircraft are involved.

The computational experiments were based on the same real traffic data in the case study described in Section 4.6. To simulate the future increasing traffic volume, the flight plans of the 31 aircraft from the traffic data were duplicated flight by flight, so that the number of aircraft that need to traverse the sector is gradually increased from 31 aircraft to 94 in total. The parameter settings of the algorithm remained the same as those in Section 4.6. The results of the experiments are summarized in Table 4.3.

The relationship between run time and the number of aircraft is plotted in Figure 4.13, which indicates that MIDA's run time increases roughly in a linear manner with the increasing number of aircraft. MIDA displays good scalability with the predicted increasing traffic volume. Even

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

Table 4.3 Results of computational experiments with increasing number of aircraft

<i>Number of aircraft</i>	<i>Min separ. (nm)</i>	<i>Dur. in 3*s (min)</i>	<i>Run time (s)</i>
31	11	0.53	1.03
34	11	1.53	2.19
37	11	1.53	3.28
40	11	2.70	5.50
43	11	2.71	8.45
46	11	2.71	10.58
49	11	2.71	12.42
52	11	2.71	14.05
55	7	3.29	15.85
58	7	3.52	19.24
61	6	8.04	21.71
64	6	8.04	24.35
67	6	15.87	28.21
70	6	13.86	36.75
73	6	17.12	36.50
76	6	20.58	38.69
79	6	21.06	41.55
82	6	21.13	43.16
85	6	23.91	48.29
88	6	35.31	49.19
91	6	36.83	53.96
94	6	39.25	61.98

4.7 Computational Experiments for Future Traffic Demand

for the largest computational case, MIDA finishes in about 1 minute. Remarkably, MIDA's run time can be further reduced if it is run on more powerful hardware for real industrial applications. Moreover, MIDA, as a distributed algorithm, can be implemented by parallel computation: in each iteration, the subproblem of each aircraft is independent of each other, so subproblems can be solved in parallel to shorten run time rather than in serial, as in the current implementation.

Figure 4.13 Run time with increasing number of aircraft

In addition, safety separations among aircraft are still strictly maintained, even under the more congested traffic situations. Figure 4.14 shows the decreasing minimum separation in terms of the increasing number of aircraft. Because intensity weight ω_2 is fixed relatively high at 100, in the experiments at the beginning, the minimum separation is maintained at 11 nm, more than twice the safety distance. However, with increases in populating aircraft,

Coordinated Multi-aircraft 4D Trajectory Planning Considering Buffer Safety Distance and Fuel Optimization

the minimum separation quickly drops. This proves that, although a large intensity weight encourages aircraft separation, the intensity term in the overall cost only serves as a kind of soft constraint. When traffic volume increases, the computed 4D trajectories by MIDA will automatically reserve a lower buffer safety distance to accommodate a higher traffic volume. Table 4.3 also shows that the total duration for which aircraft separation is less than three times the safety distance is substantially increased when there are more aircraft. In other words, aircraft fly closer to each other to make more airspace for the additional aircraft.

Figure 4.14 Minimum separation with increasing number of aircraft

Based on the computational results, it can be concluded that MIDA, as a tool for online 4D trajectory planning, will be scalable with the rapidly increasing traffic volume predicted for the upcoming decades. More significantly, from a managerial perspective, the computational experiments reveal the benefits that 4D Trajectory Based Operations will bring to Air Traffic

Control: increased airspace capacity, decreased air traffic controller workloads, and assured air safety. The shift from clearance-based to trajectory-based control will play a vital role in the future Air Traffic Management system which will be developed and improved continuously to tackle the challenges of future air traffic.

4.8 Summary

In this chapter, we considered a coordinated multi-aircraft 4D trajectory planning problem in the context of 4D Trajectory Based Operations. The problem takes into account fuel, delay, and intensity, which is mitigated by buffer safety distance, with possible aircraft maneuvers including speed adjustments and flight level changes. To solve the problem, using a discrete-time-discrete-space representation, we modeled 4D trajectories as sequences of 4D legs, and the impact of buffer safety distance was discussed and modeled. Because the minimum distance between two 4D legs can be computed analytically, conflict-free of 4D trajectories is guaranteed at any time, not only at sampled time instances. We then formulated the problem as a pure-strategy game, and developed MIDA to find an equilibrium at which no aircraft can unilaterally improve further. MIDA iterates by solving subproblems that generate single-aircraft optimal 4D trajectories efficiently, so it does not need all possible 4D trajectories as input and select from them. Proofs of the existence of the equilibrium and the convergence of the algorithm were given. A case study based on real air traffic data shows that MIDA is able to solve 4D trajectories for online application with an estimated 16.7% reduction in monetary cost and allocate abundant buffer safety distance at the minimum separation point, which results in less intense air traffic situations. Comprehensive computational experiments were carried out and MIDA's scalability was verified for the predicted doubled and tripled future traffic loads.

Chapter 5

Conclusions and Future Work

In this chapter, the key results and contributions of this dissertation are summarized. Possible directions to further advance the research of this dissertation are discussed.

5.1 Conclusions

With the advancement of the global economy, more and more people choose to travel by air. In spite of the convenience brought by air travel, with the rapid growth of air traffic demand, the inefficiency of the Air Transport System has become a salient problem. Flight delays, cancellations, and air traffic congestion cost customers and airlines a large amount of money every year. In this dissertation, we addressed the inefficiency of the Air Transport System from two perspectives, namely, airspace users (airlines) and ATM service providers (air traffic controllers). Accordingly, two critical problems were investigated at the operational level of the ATS, the aircraft recovery problem and 4D trajectory planning.

From the perspective of airspace users, we considered the aircraft recovery problem with airport capacity constraints and maintenance flexibility. Disruptions are almost inevitable in every airline's daily operations, and they have a considerable impact on the airline industry. There exists a particular need to develop computational tools for airlines to manage disruptions

Conclusions and Future Work

swiftly in real life. After an airline's original flight schedule is disrupted, the problem is how to re-schedule flights and re-assign aircraft in real time with minimized recovery cost in compliance with various operational constraints. A column generation framework was proposed to solve the problem with a provable optimality gap. The framework comprises a master problem for selecting routes for aircraft and subproblems for generating the routes. Airport capacity and maintenance flexibility are explicitly considered in the framework. Flight delays are continuous decision variables and are optimized accurately in the subproblems. The computational study based on real airline operation data shows that the master problem gives a very tight linear relaxation with small, often zero, optimality gap. Large-scale problems can be solved within 5 minutes and it was validated that the run time could be further shortened by parallelizing subproblems on more powerful computing machines.

The major contributions of the aircraft recovery problem part of this dissertation are as follows. First, we proposed an optimization approach to solving the aircraft recovery problem with airport capacity constraints and maintenance flexibility. Optimal or near-optimal solutions with optimality gaps can be obtained within a short computation time. Airport capacity is explicitly modeled in the master problem and swappable planned maintenances can be directly incorporated in the subproblems. Second, operational insights were derived for maintenance flexibility. In this model, aircraft can swap planned maintenances so as to adjust the time and place where the aircraft are maintained. Case studies show that swapping planned maintenances may bring a reduction in recovery cost by about 20% to 60%, depending on the problem instance. Furthermore, the decreasing marginal value of airport slot quotas was discovered through computational experiments. Third, regarding the modeling and algorithmic part of this work, instead of being modeled by discrete flight copies blindly and roughly, flight delays are computed adaptively and precisely as continuous variables. Compared to the discrete-delay method, the continuous-delay model improves the accuracy of the optimized recovery cost. In one test scenario, the accuracy is improved by up to 37.74%. In addition, we implemented the

column generation framework by multi-thread programming, which shortens the run time of the largest scenario by 67.76%.

From the perspective of ATM service providers, we considered the coordinated multi-aircraft 4D trajectory planning problem. With the development of advanced satellite-based communication, navigation and surveillance technologies, 4D Trajectory Based Operations has become a promising solution for the future ATM system. The core of 4D TBO is to plan 4D trajectories for aircraft traversing an ATC sector. The planned 4D trajectories need to specify each aircraft's position at any time, ensuring conflict-free operation and reducing fuel and delay costs, with possible aircraft maneuvers such as speed adjustment and flight level change. In contrast to much of the published literature, the impact of buffer safety distance is also under consideration in this dissertation, and deconfliction is guaranteed at any given time, not only at discrete time instances. We formulated the 4D trajectory planning problem as a pure-strategy game in which individual aircraft are players and all possible 4D trajectories are strategies. An efficient maximum improvement distributed algorithm was developed to find the equilibrium at which no aircraft can unilaterally improve further. Proofs of the existence of the equilibrium and the convergence of the algorithm are given. A case study based on real air traffic data shows that the algorithm is able to solve 4D trajectories for an online application with an estimated 16.7% reduction in monetary costs and allocate an even larger buffer safety distance at the minimum separation point. The scalability of the algorithm was verified by computational experiments for projected doubled and tripled future traffic loads.

The key contributions of the 4D trajectory planning part of this dissertation are as follows. First, a coordination mechanism was proposed and modeled as a pure-strategy game for online 4D trajectory planning among aircraft. As a pioneering work, the tradeoff between operating costs and buffer safety distance can be decided by an automatic decision support tool. Second, a maximum improvement distributed algorithm was carefully designed that decomposes multi-aircraft problems into single-aircraft problems, with no need to enumerate all possible 4D

Conclusions and Future Work

trajectories in advance. Its convergence to a Nash equilibrium was proved and its applicability and scalability were demonstrated by comprehensive computational experiments. Third, based on discrete airspace and discrete timeline representation, 4D trajectories are modeled as piecewise linear routes on time-space network. Hence, a fast dynamic-programming algorithm was developed for the single-aircraft problem, jointly optimizing fuel consumption, arrival delay, and buffer safety distance. Finally, from a managerial point of view, by conducting computational experiments using real traffic data and practical cost factors, the tradeoff between buffer safety distance and other operating costs were analyzed quantitatively.

In conclusion, this dissertation explored the methods and technologies to improve the efficiency of the Air Transport System in response to the imminent challenges caused by the ever-growing air traffic volume and continuously occurring operational disruptions. Two seminal problems, namely, the aircraft recovery problem and 4D trajectory planning, were studied from the two perspectives of the Air Transport System. Both problems were formulated as special routing problems, and because of the large computational scale of the two problems encountered in real life, ad hoc route generation algorithms were designed that exploit decomposition techniques. As airlines and ATM service providers are the two indispensable stakeholders of the ATS, this dissertation makes enlightening contributions to enhancing the operation of civil aviation from an optimization point of view. In general, the route generation algorithms proposed in this dissertation also provide inspirations for other real-life large-scale problems in logistics systems, such as railway transportation, maritime transportation, and most recently, unmanned aerial vehicle (UAV) transportation.

5.2 Future Work

For the aircraft recovery problem, some practical features are to be added to the model. First, multiple types of aircraft can be included, because in the subproblem, it is not difficult to check whether a flight is compatible with the aircraft's type. Second, as already discussed, conditional

turn time depending on flight type (e.g., domestic or international) or aircraft type could be accommodated by the current framework, which is built on a connection network. Third, to further accelerate the computational speed, parallel computing implemented by networked high-performance computers is also a promising direction, although it requires more advanced knowledge and skills in computer networking and programming. Moreover, from a broader point of view, as the ARP is a part of the Airline Disruptions Management, we will integrate the aircraft recovery problem with passenger itinerary recovery in future. Models considering maintenance capacity at airports are also under consideration.

For the 4D trajectory planning, the work can be further enriched in the following ways. First, in the work, assumptions are adopted. For example, aircraft are regarded as points. Aircraft trajectories are assumed to be piecewise linear. Timeline is discretized to time instances and airways are discretized to waypoints. We assume that, in each discrete time step, aircraft fly from one waypoint to another in constant speed. Aircraft can only change speed among several options at waypoints. Although the assumptions greatly simplify the computation and make online multi-aircraft 4D trajectory planning tractable, some details of the real problem are lost. For future work, more detailed aircraft dynamics shall be included, or the model will be built on a more accurate continuous-time-discrete-space framework. Second, receding horizon planning will be implemented to extend the current algorithm framework for solving problems with a longer planning horizon. Third, weather, as a critical factor in aviation, should be included in the model, possibly by modeling its impact on 4D legs directly. For instance, if adverse weather is too severe, affected 4D legs can be blocked on the time-space graph. Furthermore, the current study assumes the conformity of aircraft. In reality, deviation between the planned 4D trajectories and actual flown ones is inevitable. The positions of aircraft may be modeled using probability. Moreover, as discussed in the end of Section 4.5.2, MIDA does not guarantee to find the equilibrium which corresponds to the global optimal because the problem is "non-convex" and multiple local optimal solutions exist. In this regard, meta-heuristics

Conclusions and Future Work

such as genetic algorithms or simulated annealing may be combined to enhance the algorithm. Multiple randomized initialization shall also be tested.

Last but not least, because the current framework is for a single sector, the coordination of 4D trajectory planning among multiple neighboring sectors is of great interest, both for theoretical research and for practical implementation. The problem can be formulated by a two-stage approach. In the first stage, individual sector makes its own 4D trajectory planning. In this stage, the input is aircraft's planned entry/exit waypoints and times, and output is the optimized 4D trajectories (including the optimized entry/exit waypoints and times). Since 4D trajectories are optimized individually by sectors, there might be conflicts on neighboring sectors' borders. In the second stage, given the optimization results of the first stage, the optimized entry/exit waypoints and times will be modified and input back to the problems in the first stage for re-optimization. The process repeats until no conflict exists between neighbouring sectors. By and large, it is about coordination among multiple sectors. The difficulty is to devise an algorithm for the second stage problem. The algorithm must ensure convergence and effectiveness.

In terms of the entire Air Transport System, although airspace users and ATM service providers were both studied in this dissertation, the possibility of interaction and collaboration between them was not considered fully. For instance, in the 4D trajectory planning problem, the air traffic controllers may explicitly encourage aircraft to choose slightly more expensive but less congested trajectories, so equilibrium could be reached more quickly. Furthermore, airports, the other stakeholder of the ATS, were not researched in detail in this dissertation, although airport capacity is considered in the aircraft recovery problem. The efficacious integration of airspace users, airports, and ATM service providers is of great significance for further enhancing the efficiency and reliability of the Air Transport System as a whole.

Publications

Xiongwen Qian, Jianfeng Mao, Chun-Hsien Chen, Songlin Chen, Changpeng Yang (2017). *"Coordinated Multi-aircraft 4D Trajectories Planning Considering Buffer Safety Distance and Fuel Consumption Optimization via Pure-strategy Game"*, Transportation Research Part C: Emerging Technologies, 81: 18-35.

Xiongwen Qian, Zhe Liang, Lei Zhou (2017). *"Solving Aircraft Recovery Problem with Airport Capacity Constraints and Maintenance Flexibility"*, in preparation.

Changpeng Yang, Jianfeng Mao, **Xiongwen Qian**, Peng Wei (2017). *"Designing Robust Air Transportation Network via Minimizing Total Effective Resistance"*, IEEE Transactions on Intelligent Transportation Systems, under review.

References

- Abdelghany, K. F., Abdelghany, A. F., and Ekollu, G. (2008). An integrated decision support tool for airlines schedule recovery during irregular operations. *European Journal of Operational Research*, 185(2):825–848.
- Airlines for America (2017). U.S. passenger carrier delay costs. <http://airlines.org/data/per-minute-cost-of-delays-to-u-s-airlines/>. Accessed on 30 December 2017.
- Andersson, T. and Värbrand, P. (2004). The flight perturbation problem. *Transportation Planning and Technology*, 27(2):91–117.
- Argüello, M. F. (1997). *Framework for exact solutions and heuristics for approximate solutions to airlines' irregular operations control aircraft routing problem*. PhD thesis, The University of Texas at Austin.
- Argüello, M. F., Bard, J. F., and Yu, G. (1997). A grasp for aircraft routing in response to groundings and delays. *Journal of Combinatorial Optimization*, 1(3):211–228.
- Averty, P., Collet, C., Dittmar, A., Athenes, S., and Vernet-Maury, E. (2004). Mental workload in air traffic control: an index constructed from field tests. *Aviation, Space, and Environmental Medicine*, 75(4):333–341.
- Ball, M. O., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A. R., Peterson, E., Sherry, L., Trani, A., and Zou, B. (2010). Total delay impact study: A comprehensive assessment of the costs and impacts of flight delay in the united states. Technical report, Federal Aviation Administration.
- Barnhart, C., Boland, N. L., Clarke, L. W., Johnson, E. L., Nemhauser, G. L., and Shenoi, R. G. (1998). Flight string models for aircraft fleet and routing. *Transportation Science*, 32(3):208–220.
- BBC (2011). Thousands hit by volcanic ash cloud flight delays. <http://www.bbc.com/news/uk-13513981>. Accessed on 28 April 2017.
- Belobaba, P., Odoni, A., and Barnhart, C. (2009). *The Global Airline Industry*. John Wiley & Sons.
- Bierlaire, M., Eggenberg, N., and Salani, M. (2007). Column generation methods for disrupted airline schedules. Technical report, Ecole Polytechnique Federale de Lausanne.
- Bisaillon, S., Cordeau, J.-F., Laporte, G., and Pasin, F. (2011). A large neighbourhood search heuristic for the aircraft and passenger recovery problem. *4OR*, 9(2):139–157.

References

- Bonami, P., Olivares, A., Soler, M., and Staffetti, E. (2013). Multiphase mixed-integer optimal control approach to aircraft trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 36(5):1267–1277.
- Bratu, S. and Barnhart, C. (2006). Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, 9(3):279–298.
- Byung-Wook, W. (1995). A differential game approach to the dynamic mixed behavior traffic network equilibrium problem. *European Journal of Operational Research*, 83(1):117–136.
- Campbell, S. E. (2012). Multiscale path optimization for the reduced environmental impact of air transportation. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1327–1337.
- Cao, J. and Kanafani, A. (1997a). Real-time decision support for integration of airline flight cancellations and delays part I: mathematical formulation. *Transportation Planning and Technology*, 20(3):183–199.
- Cao, J.-M. and Kanafani, A. (1997b). Real-time decision support for integration of airline flight cancellations and delays part II: algorithm and computational experiments. *Transportation Planning and Technology*, 20(3):201–217.
- Chen, O. and Ben-Akiva, M. (1998). Game-theoretic formulations of interaction between dynamic traffic control and dynamic traffic assignment. *Transportation Research Record: Journal of the Transportation Research Board*, 1617:179–188.
- Civil Aviation Authority of China (2014). China Aeronautical Information Publication (in Chinese). Technical report.
- Clausen, J., Larsen, A., Larsen, J., and Rezanova, N. J. (2010). Disruption management in the airline industry—concepts, models and methods. *Computers & Operations Research*, 37(5):809–821.
- Cook, G. N. and Billig, B. (2017). *Airline Operations and Management*. Routledge Publishing.
- Delahaye, D., Puechmorel, S., Tsiotras, P., and Feron, E. (2014). Mathematical models for aircraft trajectory design: A survey. In *Air Traffic Management and Systems*, volume 290 of *Lecture Notes in Electrical Engineering*, pages 205–247. Springer.
- Dell’Olmo, P. and Lulli, G. (2003). A new hierarchical architecture for air traffic management: Optimisation of airway capacity in a free flight scenario. *European Journal of Operational Research*, 144(1):179–193.
- Department of Transportation (2016). Airline on-time statistics and delay causes. https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp?pn=1. Accessed on 25 April Year 2017.
- Desaulniers, G., Desrosiers, J., and Solomon, M. M. (2005). *Column Generation*. Springer.
- Djokic, J., Lorenz, B., and Fricke, H. (2010). Air traffic control complexity as workload driver. *Transportation Research Part C: Emerging Technologies*, 18(6):930–936.

- Dwyer, C. (2017). On day 3 of cancellation chaos, British Airways edges closer to recovery. <http://www.npr.org/sections/thetwo-way/2017/05/29/530587512/on-day-3-of-cancellation-chaos-british-airways-edges-closer-to-recovery>. Accessed on 21 June 2017.
- Eggenberg, N., Salani, M., and Bierlaire, M. (2010). Constraint-specific recovery network for solving airline recovery problems. *Computers & Operations Research*, 37(6):1014–1026.
- EUROCONTROL (2012). European ATM Master Plan. Technical report.
- EUROCONTROL (2014). Second i4d test flight crosses muac airspace successfully. <http://www.eurocontrol.int/news/second-i4d-test-flight-crosses-muac-airspace-successfully>. Accessed on 20 Mar 2015.
- EUROCONTROL (2015). European ATM Master Plan. Technical report.
- EUROCONTROL (2016). BADA Calculation Tool. <https://badaext.eurocontrol.fr/home>. Accessed on 25 September 2016.
- FAA (2013). NextGen Implementation Plan. Technical report.
- FAA (2014). Improving Air Traffic Management Together. <http://cdm.fly.faa.gov/>. Accessed on 30 April 2017.
- FAA (2016). Trajectory Based Operations (TBO) Animated Storyboard. http://www.faa.gov/about/office_org/headquarters_offices/ang/offices/tc/library/v&vsummit/v&vsummit2016/presentations/9%20use%20of%20storyboarding%20for%20trajectory%20based%20operations.ppsx. Accessed on 20 February 2017.
- Fisk, C. (1980). Some developments in equilibrium traffic assignment. *Transportation Research Part B: Methodological*, 14(3):243–255.
- Franco, A. and Rivas, D. (2015). Optimization of multiphase aircraft trajectories using hybrid optimal control. *Journal of Guidance, Control, and Dynamics*, 38(3):452–467.
- Gao, C., Johnson, E., and Smith, B. (2009). Integrated airline fleet and crew robust planning. *Transportation Science*, 43(1):2–16.
- Gardi, A., Sabatini, R., and Ramasamy, S. (2016). Multi-objective optimisation of aircraft flight trajectories in the ATM and avionics context. *Progress in Aerospace Sciences*, 83:1–36.
- ICAO (2007). Procedures for Air Navigation Services - Air Traffic Management. Technical report.
- ICAO (2012). Report of the Asia/Pacific Area Traffic Forecasting Group. Technical report.
- ICAO (2013). Slot allocation. Worldwide Air Transport Conference, Montréal, 18 to 22 March 2013.
- Jarrah, A. I. Z., Yu, G., Krishnamurthy, N., and Rakshit, A. (1993). A decision support framework for airline flight cancellations and delays. *Transportation Science*, 27(3):266–280.

References

- Johnson, K. O., Hsiao, S. S., and Yoshioka, T. (2002). Neural coding and the basic law of psychophysics. *The Neuroscientist*, 8(2):111–121.
- Jozefowicz, N., Mancel, C., and Mora-Camino, F. (2013). A heuristic approach based on shortest path problems for integrated flight, aircraft, and passenger rescheduling under disruptions. *Journal of the Operational Research Society*, 64(3):384–395.
- Kamgarpour, M., Soler, M., Tomlin, C. J., Olivares, A., and Lygeros, J. (2011). Hybrid optimal control for aircraft trajectory design with a variable sequence of modes. In *IFAC World Congress*, volume 18, pages 7238–7243.
- Kügler, D. (2016). Course M6932 "The Fundamentals of Air Traffic Control". Taught in Nanyang Technological University on 13 July 2016.
- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358.
- Lettovsky, L. (1997). *Airline operations recovery: an optimization approach*. PhD thesis, Georgia Institute of Technology.
- Liang, Z. and Chaovalitwongse, W. A. (2013). A network-based model for the integrated weekly aircraft maintenance routing and fleet assignment problem. *Transportation Science*, 47(4):493–507.
- Liang, Z., Chaovalitwongse, W. A., and Elsayed, E. A. (2014). Sequence assignment model for the flight conflict resolution problem. *Transportation Science*, 48(3):334–350.
- Liang, Z., Feng, Y., Zhang, X., Wu, T., and Chaovalitwongse, W. A. (2015). Robust weekly aircraft maintenance routing problem and the extension to the tail assignment problem. *Transportation Research Part B: Methodological*, 78:238–259.
- Loft, S., Sanderson, P., Neal, A., and Mooij, M. (2007). Modeling and predicting mental workload in en route air traffic control: Critical review and broader implications. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 49(3):376–399.
- Maher, S. J. (2016). Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, 50(1):216–239.
- McCartney, S. (2007). Stuck in a plane on a runway: How nightmare delays happen. http://journalstar.com/business/stuck-in-a-plane-on-a-runway-how-nightmare-delays/article_e41b4e1e-a387-5425-8e73-2ede3603a74d.html. Accessed on 14 Feb 2017.
- Nagaoka, S. and Brown, M. (2015). Constructing an index of difficulty for air traffic control using proximity parameters. *Procedia Engineering*, 99:253–258.
- Petersen, J. D., Sölveling, G., Clarke, J.-P., Johnson, E. L., and Shebalov, S. (2012). An optimization approach to airline integrated recovery. *Transportation Science*, 46(4):482–500.
- Prats, X., Puig, V., Quevedo, J., and Nejjari, F. (2010). Multi-objective optimisation for aircraft departure trajectories minimising noise annoyance. *Transportation Research Part C: Emerging Technologies*, 18(6):975–989.

- Richards, A. and How, J. P. (2002). Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proceedings of the 2002 American Control Conference*, volume 3, pages 1936–1941 vol.3.
- Rosenberger, J. M., Johnson, E. L., and Nemhauser, G. L. (2003). Rerouting aircraft for airline recovery. *Transportation Science*, 37(4):408–421.
- Roy, K. and Tomlin, C. J. (2007). Solving the aircraft routing problem using network flow algorithms. In *Proceedings of the 2007 American Control Conference*, pages 3330–3335.
- Sabre Airline Solutions (2016). Competition in solving airline optimization problems. <http://page.sabreairlinesolutions.com/orcompetition>. Accessed on 10 June 2016.
- SESAR (2014). i4D and SESAR. <https://youtu.be/CI8y9QL6K3Y>. Accessed on 31 December 2017.
- Shao, S., Sherali, H. D., and Haouari, M. (2017). A novel model and decomposition approach for the integrated airline fleet assignment, aircraft routing, and crew pairing problem. *Transportation Science*, 51(1):233–249.
- Sherali, H. D., Staats, R. W., and Trani, A. A. (2003). An airspace planning and collaborative decision-making model: Part I—probabilistic conflicts, workload, and equity considerations. *Transportation Science*, 37(4):434–456.
- Sherali, H. D., Staats, R. W., and Trani, A. A. (2006). An airspace planning and collaborative decision-making model: Part II—cost model, data considerations, and computations. *Transportation Science*, 40(2):147–164.
- Sinclair, K., Cordeau, J.-F., and Laporte, G. (2014). Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. *European Journal of Operational Research*, 233(1):234–245.
- Sinclair, K., Cordeau, J.-F., and Laporte, G. (2016). A column generation post-optimization heuristic for the integrated aircraft and passenger recovery problem. *Computers & Operations Research*, 65:42–52.
- Soler, M., Zou, B., and Hansen, M. (2014). Flight trajectory design in the presence of contrails: Application of a multiphase mixed-integer optimal control approach. *Transportation Research Part C: Emerging Technologies*, 48:172–194.
- Sridhar, B., Ng, H., and Chen, N. (2011). Aircraft trajectory optimization and contrails avoidance in the presence of winds. *Journal of Guidance, Control, and Dynamics*, 34(5):1577–1584.
- Teodorović, D. and Guberinić, S. (1984). Optimal dispatching strategy on an airline network after a schedule perturbation. *European Journal of Operational Research*, 15(2):178–182.
- Thengvall, B. G., Bard, J. F., and Yu, G. (2000). Balancing user preferences for aircraft schedule recovery during irregular operations. *IIE Transactions*, 32(3):181–193.
- Thengvall, B. G., Bard, J. F., and Yu, G. (2003). A bundle algorithm approach for the aircraft schedule recovery problem during hub closures. *Transportation Science*, 37(4):392–407.

References

- Thengvall, B. G., Yu, G., and Bard, J. F. (2001). Multiple fleet aircraft schedule recovery following hub closures. *Transportation Research Part A: Policy and Practice*, 35(4):289–308.
- Thomson Reuters (2017a). British Airways CEO puts cost of recent IT outage at 80 million pounds. <http://www.businessinsider.com/r-british-airways-ceo-puts-cost-of-recent-it-outage-at-80-million-pounds-2017-6>.
- Thomson Reuters (2017b). British Airways had to cancel 60 percent of flights after IT outage. <https://www.usnews.com/news/technology/articles/2017-06-06/british-airways-had-to-cancel-60-percent-of-flights-after-it-outage>. Accessed on 21 June 2017.
- Wei, Z., Kamgarpour, M., Dengfeng, S., and Tomlin, C. J. (2012). A hierarchical flight planning framework for air traffic management. *Proceedings of the IEEE*, 100(1):179–194.
- Yan, S. and Lin, C.-G. (1997). Airline scheduling for the temporary closure of airports. *Transportation Science*, 31(1):72–82.
- Zhang, C. (2017). Two-stage heuristic algorithm for aircraft recovery problem. *Discrete Dynamics in Nature and Society*, 2017. Article ID 9575719.
- Zhang, D., Yu, C., Desai, J., and Lau, H. Y. K. H. (2016). A math-heuristic algorithm for the integrated air service recovery. *Transportation Research Part B: Methodological*, 84:211–236.
- Zou, B., Buxi, G., and Hansen, M. (2013). Optimal 4-D aircraft trajectories in a contrail-sensitive environment. *Networks and Spatial Economics*, pages 1–32.

Appendix A

Appendix for Chapter 3

The algorithms for the subproblems considering planned preventive maintenances in Section 3.4.4.3 are as follows.

Algorithm A.1 Process flight-flight arc (i, j)

```

for each label  $\langle \bar{\beta}_i, \beta_i^d, u_i^h, u_i^o, u_i^v \rangle$  in node  $i$ 's label set  $B_i$  do
  if flight  $j$ 's scheduled departure time  $t_j^s <$ 
    flight  $i$ 's scheduled arrival time  $t_i^e$  + flight  $i$ 's delay  $\beta_i^d$  + TurnTime then
    Set flight  $j$ 's delay  $\beta_j^d = t_i^e + \beta_i^d + \textit{TurnTime} - t_j^s$ .
  else
    Set  $\beta_j^d = 0$ .
  end if
  if  $\beta_j^d > \textit{MaximumDelay}$  then
    continue loop
  end if
  if flight  $j$ 's arrival time delayed by  $\beta_j^d >$  aircraft  $c$ 's end available time then
    continue loop
  end if
  if  $u_i^h + t_j^e - t_j^s > \textit{MaximumFlyingHour}$  then
    continue loop
  end if
  if  $u_i^o + 1 > \textit{MaximumCycle}$  then
    continue loop
  end if
  if  $u_i^v + (t_j^e + \beta_j^d) - (t_i^e + \beta_i^d) > \textit{MaximumInterval}$  then
    continue loop
  end if
  if flight  $j$ 's planned aircraft is aircraft  $c$  then
    Set  $\bar{\beta}_j = \bar{\beta}_i + \beta_j^d - \pi_j$ .
  else
    Set  $\bar{\beta}_j = \bar{\beta}_i + \beta_j^d - \pi_j + \textit{FlightSwapCost}$ .
  end if
  Set  $u_j^h = u_i^h + t_j^e - t_j^s$ .
  Set  $u_j^o = u_i^o + 1$ .
  Set  $u_j^v = u_i^v + (t_j^e + \beta_j^d) - (t_i^e + \beta_i^d)$ .
  if label  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$  is not dominated by any label in node  $j$ 's label set  $B_j$  then
    Insert  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$  to  $B_j$ .
    Delete the labels which are dominated by  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$  in  $B_j$ .
    Mark  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$ 's predecessor as  $\langle \bar{\beta}_i, \beta_i^d, u_i^h, u_i^o, u_i^v \rangle$ .
  end if
end for

```

Algorithm A.2 Process flight-maintenance arc (i, j)

```
for each label  $\langle \bar{\beta}_i, \beta_i^d, u_i^h, u_i^o, u_i^v \rangle$  in node  $i$ 's label set  $B_i$  do
  if maintenance  $j$  is not compatible with aircraft  $c$  then
    break loop
  end if
  if maintenance  $j$ 's scheduled departure time  $t_j^s <$ 
    flight  $i$ 's scheduled arrival time  $t_i^e +$  flight  $i$ 's delay  $\beta_i^d$  then
    continue loop
  end if
  if maintenance  $j$ 's planned aircraft is aircraft  $c$  then
    Set  $\bar{\beta}_j = \bar{\beta}_i - \pi_j$ .
  else
    Set  $\bar{\beta}_j = \bar{\beta}_i - \pi_j + \text{MaintSwapCost}$ .
  end if
  Set  $\beta_j^d = 0$ .
  Set  $u_j^h = 0$ ,  $u_j^o = 0$ , and  $u_j^v = 0$ .
  if label  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$  is not dominated by any label in node  $j$ 's label set  $B_j$  then
    Insert  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$  to  $B_j$ .
    Delete the labels which are dominated by  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$ .
    Mark  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$ 's predecessor as  $\langle \bar{\beta}_i, \beta_i^d, u_i^h, u_i^o, u_i^v \rangle$ .
  end if
end for
```

Algorithm A.3 Process maintenance-flight arc (i, j)

```

for each label  $\langle \bar{\beta}_i, \beta_i^d, u_i^h, u_i^o, u_i^v \rangle$  in node  $i$ 's label set  $B_i$  do
  if flight  $j$ 's scheduled departure time  $t_j^s <$ 
    maintenance  $i$ 's scheduled end time  $t_i^e$  then
    Set flight  $j$ 's delay  $\beta_j^d = t_i^e - t_j^s$ .
  else
    Set flight  $j$ 's delay  $\beta_j^d = 0$ .
  end if
  if  $\beta_j^d > \text{MaximumDelay}$  then
    continue loop
  end if
  if flight  $j$ 's arrival time delayed by  $\beta_j^d >$  aircraft  $c$ 's end available time then
    continue loop
  end if
  if  $u_i^h + t_j^e - t_j^s > \text{MaximumFlyingHour}$  then
    continue loop
  end if
  if  $u_i^o + 1 > \text{MaximumCycle}$  then
    continue loop
  end if
  if  $u_i^v + (t_j^e + \beta_j^d) - (t_i^e + \beta_i^d) > \text{MaximumInterval}$  then
    continue loop
  end if
  if flight  $j$ 's planned aircraft is aircraft  $c$  then
    Set  $\bar{\beta}_j = \bar{\beta}_i + \beta_j^d - \pi_j$ .
  else
    Set  $\bar{\beta}_j = \bar{\beta}_i + \beta_j^d - \pi_j + \text{FlightSwapCost}$ .
  end if
  Set  $u_j^h = u_i^h + t_j^e - t_j^s$ .
  Set  $u_j^o = u_i^o + 1$ .
  Set  $u_j^v = u_i^v + (t_j^e + \beta_j^d) - (t_i^e + \beta_i^d)$ .
  if label  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$  is not dominated by any label in node  $j$ 's label set  $B_j$  then
    Insert  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$  to  $B_j$ .
    Delete the labels which are dominated by  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$  in  $B_j$ .
    Mark  $\langle \bar{\beta}_j, \beta_j^d, u_j^h, u_j^o, u_j^v \rangle$ 's predecessor as  $\langle \bar{\beta}_i, \beta_i^d, u_i^h, u_i^o, u_i^v \rangle$ .
  end if
end for

```