TACKLING AIRCRAFT ROUTING UNCERTAINTIES: ADJUSTABLE CRUISE SPEED AND FUZZY MODELLING APPROACH

ZHANG QING

PhD

The Hong Kong Polytechnic University

2023

The Hong Kong Polytechnic University

Department of Industrial and Systems Engineering

Tackling Aircraft Routing Uncertainties: Adjustable Cruise Speed and Fuzzy Modelling Approach

Zhang Qing

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

August 2022

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in this text.

_______________________________________ (Signed)

Zhang Qing _______________________________ (Name of student)

Abstract

Aircraft maintenance routing problem (AMRP) is crucial for airline planning due to its significant impact on aircraft utilization and schedule reliability. It is known that allowing flight flying time variability in aircraft re-routing can achieve improved flight connection opportunities, thus higher aircraft utilization and enhanced schedule flexibility. However, the similar impact on aircraft routing is under-explored. In this research, we develop a new AMRP model that incorporates cruise speed control to take advantage of flexibility in fight flying times. In the proposed model, each flight leg is assigned a cruise time window where several leg copies with different cruise times are placed and only one copy can be selected by one flight leg. The objective function is to minimize the sum of aircraft usage costs, idle time costs and fuel-burn related costs so that a critical trade-off between the aircraft utilization and fuel-burn related costs can be examined.

However, the combination of two intricate sets of decisions, i.e., cruise time for each flight leg and maintenance route for each aircraft, poses significant methodological challenges. To solve the problem efficiently, we first develop an improved ant colony optimization (IACO) algorithm with a new state transition mechanism and a new pheromone updating mechanism to enhance the search efficiency and precision. Then, we propose a matheuristic approach which is comprised of three components: the IACO algorithm, the set partitioning (SP) procedure and the neighborhood search (NS) procedure. The IACO algorithm serves as a route generator, populating a pool of routes with promising feasible aircraft maintenance routes. A SP model, which features the high-quality columns corresponding to the routes in the pool, is solved to produce a possible better solution. This solution is further improved by a NS procedure that iteratively solves the reduced instances to optimality.

Despite the extensive studies in the operational side of AMRP, robust AMRP (RAMRP) attracts more attention due to the prevalent and costly disruptions in operating environment. However, most studies focus on aircraft routing while the maintenance regulations are either disregarded or used as constraints. In fact, aircraft maintenance is an important source of disruption. Especially in current practice, a maintenance A-check program is divided into multiple task packages, each with

varying tasks and durations, complicating its operation and thereby increasing the disruption risk. To alleviate the impact of maintenance disruption, we first accurately assess the disruption risk for each task package using fuzzy logic approach due to the different levels of risk associated with each package. Then, based on the assessment results, a new robustness enhancement strategy is developed, of which the core idea is to identify the appropriate buffer time allocation for task packages. Besides, a robustness measurement, namely the total risk score, is proposed to construct a new RAMRP model. Finally, a matheuristic approach is developed to effectively solve the RAMRP model.

Publications Arising from the Thesis

International Journal

Zhang, Q., Chan, F. T. S., Chung, S. H., & Fu, X. W., (2022). Operational aircraft maintenance routing problem incorporating cruise speed control. *Engineering Optimization*. (Minor revision)

Zhang, Q., Chan, F. T. S., Chung, S. H., & Fu, X. W., (2022). A matheuristic approach for aircraft maintenance routing problem incorporating cruise speed control. *Expert Systems with Applications*. (Invited for re-submission)

International Conference Papers

Zhang, Q., Chan, F. T. S., Chung, S. H., & Fu, X. W., (2021). The aircraft maintenance routing problem incorporating cruise speed control. In the Conference on The Global Summit and Expo on Aerospace and Mechanical Engineering (GSEAME2021), 18-20 October 2021, Valencia, Spain. (Accepted)

Acknowledgements

Firstly, I would like to express my sincere gratitude and appreciation to my supervisor, Dr. Nick S.H. Chung. It is his patient instruction and valuable suggestion that greatly help me better understand my research work. Thanks to his professionalism and immense knowledge, my research could go in the right direction. I am really appreciated for having the opportunity to conduct my research project under the supervision of Dr. Nick S.H. Chung. Moreover, I wish to express my sincere gratitude to my co-supervisor Professor Felix T.S. Chan. His continuous guidance, generous encouragement and endless support inspire me to finish my PhD study. Besides, special thanks are given to my co-supervisor Professor FU Xiaowen.

Finally, I wish to express my special thanks to my family for their moral support and encouragement throughout my PhD period. I would like to extend my thanks to my friends for their motivation to finish this PhD.

Thank you for your support.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AMPR	Aircraft Maintenance Routing Problem
AMRP-CSC	Aircraft Maintenance Routing Problem incorporating Cruise Speed Control
ACO	Ant Colony Optimization
IACO	Improved Ant Colony Optimization
TMAPR	Tactical Aircraft Maintenance Routing Problem
OAMRP	Operational Aircraft Maintenance Routing Problem
RAMRP	Robust Aircraft Maintenance Routing Problem
SDAMRP	Single Day Aircraft Maintenance Routing Problem
FSP	Flight Scheduling Problem
FAP	Fleet Assignment Problem
CSP	Crew Scheduling Problem
CPP	Crew Pairing Problem
CRP	Crew Rostering Problem
IATA	International Air Transport Association
BTS	Bureau of Transportation Statistics
BADA	The Base of Aircraft Data
FAA	Federal Aviation Administration
SA	Simulated Annealing
GA	Genetic Algorithm
MOGA	Multi-Objective Genetic Algorithm
LOF	Lines Of Flying
CCA	Crew-Compatible Aircraft
O&D	Origin and Destination
VRPs	Vehicle Routing Problems

TSPs	Travelling Salesman Problems
IP	Integer Programming
MIP	Mix Integer Programming
ILP	Integer Linear Programming
MILP	Mix Integer Linear Programming
VNS	Variable Neighborhood Search
MDP	Markov Decision Process
OAMRPFD	OAMRP with flight delay consideration

Chapter 1 - Introduction

1.1 Research background

1.1.1 Aircraft maintenance routing problem (AMRP)

Given the initial strategic decisions, before the schedule can be actually implemented, airline planners should solve a series of problems:

1. Flight scheduling problem (FSP): this planning procedure is providing a list of flight legs, each characterized by a specified timetable. It involves two dependent phases: schedule construction and schedule evaluation. Considering the projected demand, the market share and the time slots of the available airports, schedule construction is used to generate the initial timetable. Then the draft timetable will be examined in the evaluation stage to check its feasibility, costs or other performance considerations. The information obtained in this stage will be used in the first stage to further improve the timetable. These two stages are repeated throughout the flight scheduling process until a desirable timetable can be derived.

2. Fleet assignment problem (FAP): based on the flight schedules, FAP is to handle the assignment of aircraft types to individual flight legs. Imagine a large aircraft performing a flight leg with a small number of passengers or a small aircraft covering a flight leg with high passenger demand. The first scenario will result in unnecessary operating costs while the potential revenue will be lost in the second scenario. Therefore, matching the aircraft type to passenger demand is a necessity in FAP, so that the total profit can be maximized. Three fundamental constraints are applied during the fleet assignment process: (i) one flight leg can be covered by exactly one fleet type, (ii)the flow balance should be ensured at each airport, and (iii)the number of used aircraft should not exceed the number of available aircraft.

3. Aircraft maintenance routing problem (AMRP): after flight schedules have been fleeted, AMRP is to identify the sequence of flight legs for individual aircraft in a given fleet, while adhering to the maintenance regulations mandated by the Federal Aviation Administration (FAA). There are four types of maintenance checks, each with different frequencies and durations.

i. The type A check is the first major check, mandated by the FAA. It should be operated every 65-125 flying hours, or about once a week and takes roughly 6-8 hours to complete. The type A check involves the examination of all key components such as engines and landing gear. In practice, the maximum allowable flying hours is about 35 to 40 hours, which is more stringent than the FAA regulations.

ii. The type B check should be undertaken after 300–600 flying hours and takes approximately 1–3 days. It entails performing a comprehensive visual check and lubricating all moving parts.

iii. Type C and type D checks are performed once every one to four years and take up to a month at a time. Due to the dynamic nature of the market and the relatively large intervals between the two checks, it is not necessary to consider these two checks in AMRP.

As the A check is the most frequent check and impacts on the daily scheduling most, we only consider A check in our work. When constructing the aircraft maintenance route, we must ensure these basic constraints:

i. Flight coverage constraints: each flight leg can only be operated by one aircraft and meanwhile one aircraft can only fly one flight leg.

ii. Time issue for flight connections: the two flight legs can be consecutively performed by one aircraft if and only if the connection time between the two flights is equal to or greater than the minimum turn time of this aircraft.

iii. Place issue for flight connections: if the two flight legs can be operated in sequence by the same aircraft, the arrival airport of the former flight leg must be identical to the departure airport of the latter flight leg.

iv. Maintenance regulations: there are three maintenance regulations including maximum allowable flying time, maximum allowable take-offs and maximum allowable flying days.

4. Crew scheduling problem (CSP): CSP is responsible for assigning crew members to individual aircraft with the goal of saving costs. Because of tractability issues, the CSP is broken into two stages: crew pairing problem (CPP) and crew rostering problem (CRP). The CPP is to generate a set of feasible pairings with the objective of minimizing the total cost of crew assignment while adhering to various regulations. The CRP is to assign the pairings to specific crew members so that their skills, vacations, and other requirements can be satisfied.

Due to the complexity of the planning problems, four sub-problems are solved sequentially, where the decisions made in the former problem form the input of the subsequent problem. Among these problems, AMRP is the main focus of our study, which is solved after fleet assignment. AMRP plays a crucial role in achieving an airline's profitability because it directly affects aircraft utilization. In AMRP, both inefficient flight connections and inappropriate maintenance arrangements will incur unnecessary ground time, thus having tremendously adverse impacts on aircraft utilization.

1.1.2 Aircraft maintenance routing problem incorporating cruise speed control (AMRP-CSC)

Purchasing an aircraft requires huge up-front capital investment which can be clearly seen in figure 1.1 and figure 1.2. Moreover, airlines need to cover the depreciation costs during the useful life of an aircraft. Therefore, airlines increasingly rely on aircraft leasing in their fleets (Oum et al. 2000; Chen et al. 2018) . In the past decades, the share of leased aircraft in the world's airline fleets has increased exponentially from half a percent in 1970 to 51 percent in 2021, as shown in Figure 1.3 and Table 1.1. (BTS. 2022). Among all the types of aircraft lease, the operating lease, which is adopted to acquire additional fleet capacity for a short period of time, is much more favorable for airlines (Bourjade et al. 2017). It is more operationally flexible, allowing for frequent renewal of the fleet, thereby enabling the fleet composition and size to be managed closely to the flights operations. (Oum et al. 2000). This feature allows the schedule planning decisions to provide the basis for further investigations regarding fleet composition and size. Clearly, the number of required aircraft determined for AMRP can directly impact on the number of aircraft obtained by operating lease. However, compared with other aircraft leasing alternatives and aircraft purchases, rental costs for operating leases appear to be higher in terms of monthly payments. Moreover, the rental costs increase with the increasing demand for operating leases in the market (Belobaba 2009). It means when the airlines need to expand their fleet capacity most, the rental costs will also be the highest (Oum et al. 2000). In this regard, the originally low profit margin is further shrunk. To achieve profitability, it is of great significance to decrease the number of required aircraft in long-term AMRP.

However, to maximize aircraft utilization and thus reduce the number of required aircraft in AMRP, the aircraft maintenance route is constructed with little buffer time for absorbing even minor delays, resulting in rapid delay propagation. In practice, airlines suffer from various uncertainties, such as aircraft breakdowns, crew sickness, and bad weather(Ahmed et al. 2017). Consequently, severe disruptions can be easily incurred, followed by significant delay costs, and decreased aircraft utilization. Therefore, in addition to increasing aircraft utilization, it is also an absolute necessity to improve schedule stability and flexibility so that airlines can better manage disruptions. Herein, stability means the ability of schedules that is less susceptible to disturbances while flexibility relates to the capacity of schedules to be easily repaired once disrupted (Eltoukhy et al. 2019).

Fortunately, we found that allowing variability in flight flying time (which is primarily achieved by cruise speed control) in aircraft re-routing can achieve higher aircraft utilization and enhanced schedule flexibility by improving flight connection opportunities (Gürkan et al. 2016). This impact of cruise speed control on aircraft re-routing can be explained in the following way. For two flight legs that can be operated successively by identical aircraft, a sufficient time interval between their departure times is required. It is comprised of cruise time, non-cruise time, turnaround time and idle time. By cruise speed control, we can adjust its lower bound and thereby transform the infeasible flight connections into feasible. This allows for the construction of new efficient routes and the increased possibilities of aircraft swaps, which enhances aircraft utilization and schedule flexibility.

Based on the aforementioned observations, it is valuable and important to apply cruise speed control in AMRP to decrease the number of required aircraft and improve schedule flexibility.

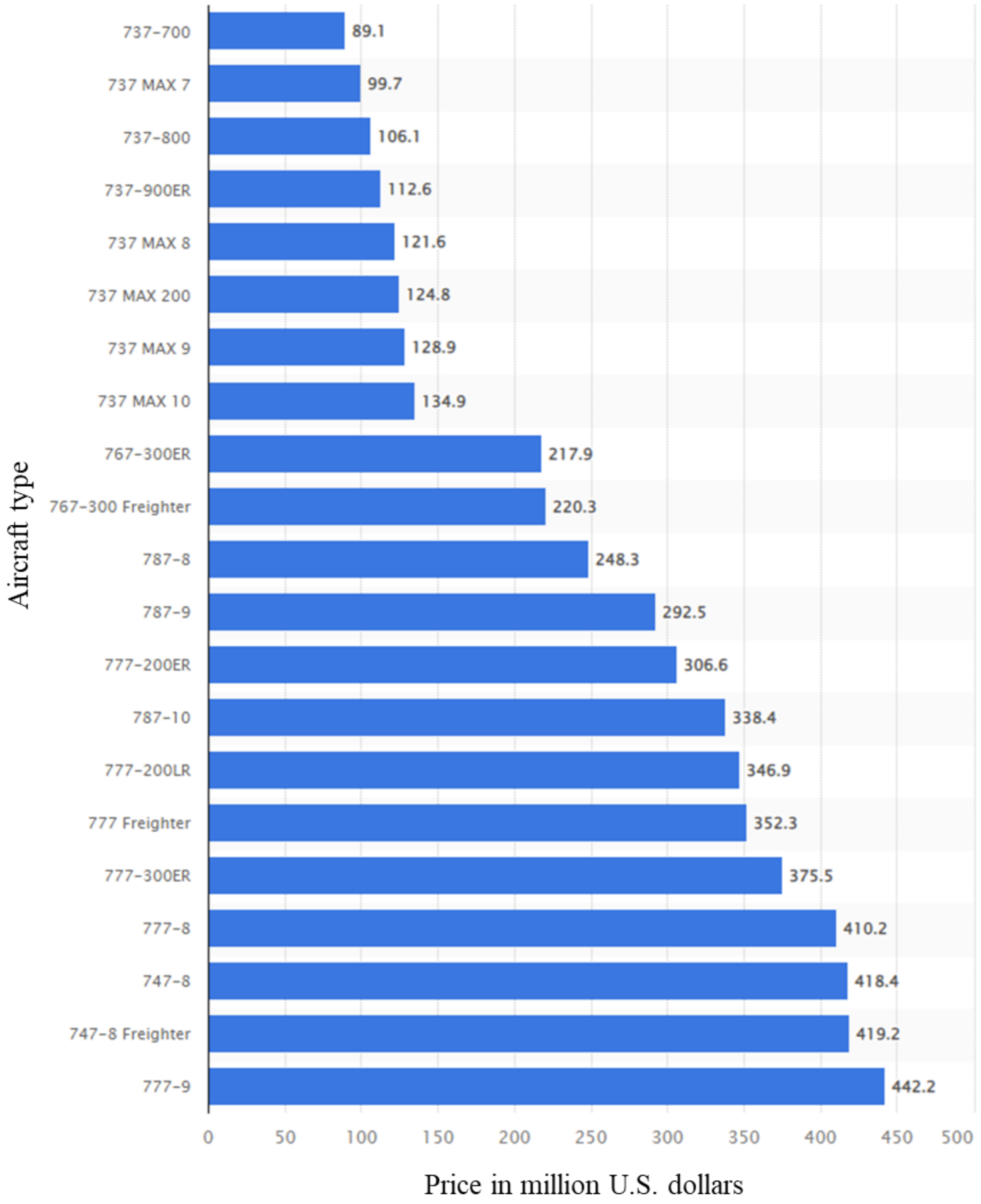


Figure 1.1: Average prices for Boeing aircraft as of March 2021, by type

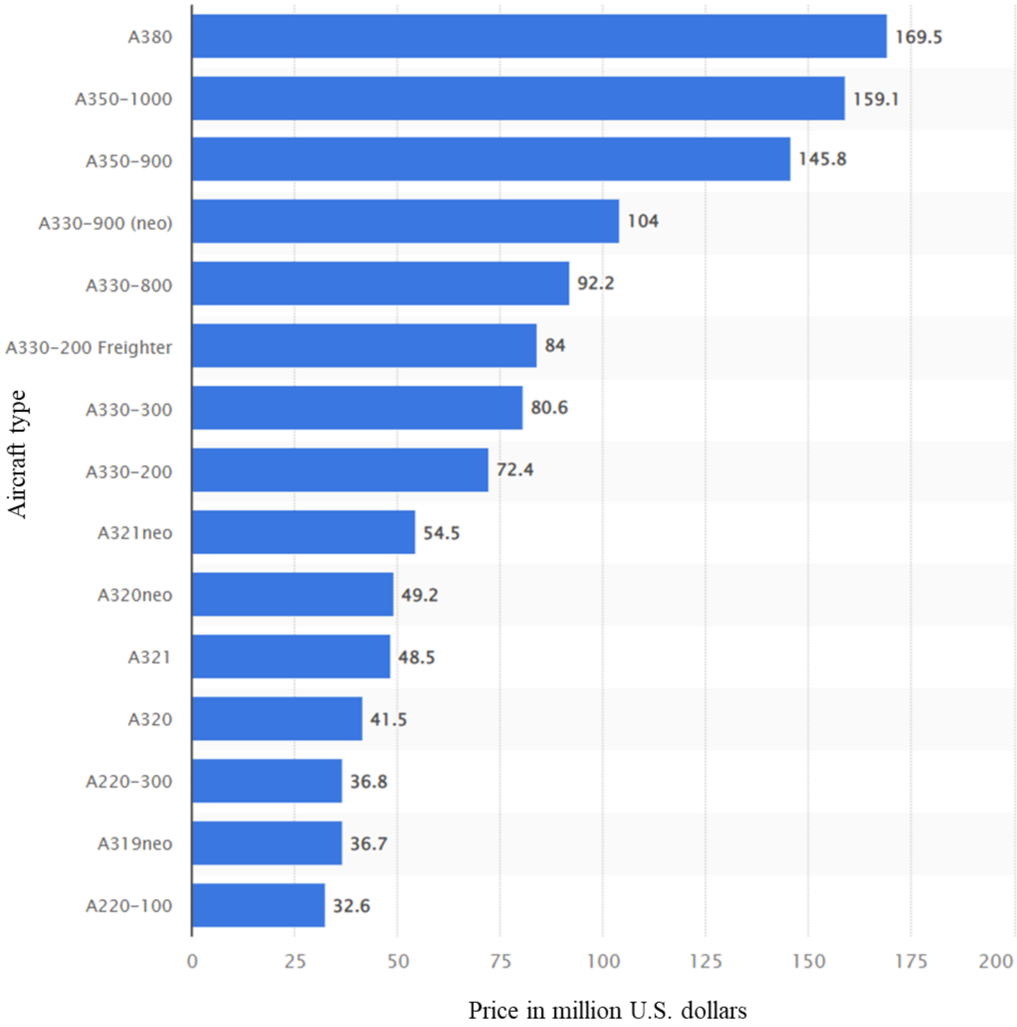


Figure 1.2: Average prices for Airbus aircraft as of March 2021, by type

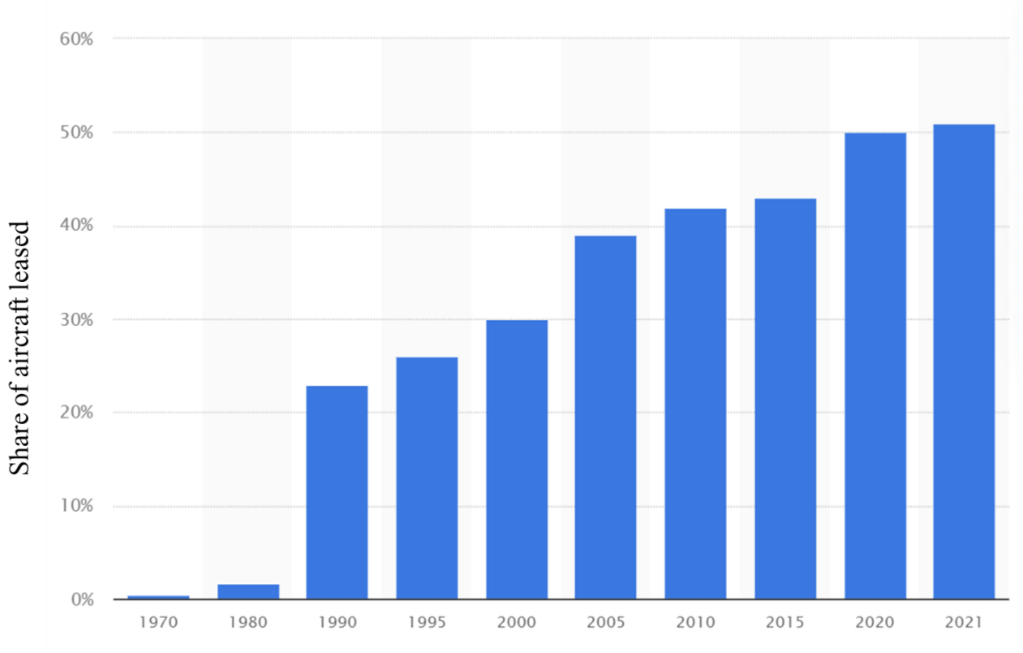


Figure 1.3: The share of leased aircraft in the world's airline fleets

Table 1.1: The share of leased aircraft in the world's airline fleets

Years	1970	1980	1990	2000	2014	2020	2021
Percentage	0.5%	1.7%	14.7%	24.7%	40.7%	50%	51%

1.1.2.1 Cruise speed control

In the very beginning, cruise speed control was widely utilized by pilots to reduce propagated delay or fuel consumption during the operation. However, the pilots cannot evaluate how their decisions on cruise speed impacts the overall network, which prevents the airlines from taking the full advantage of the cruise speed control. Recognizing this weakness in operational application of cruise speed control, many studies considered cruise speed control from the perspective of global optimization. They considered the impact of cruise speed control on the network connectivity of the flight's aircraft, crew, and passengers to downstream flights and applied it into schedule recovery problems, i.e., schedule re-optimization problems and schedule disruption recovery problems. In the studies regarding schedule re-optimization, for one thing, cruise speed control can be employed as a cost-effective alternative to idle time insertion for increasing schedule robustness by maintaining the passenger connection service levels. For another, it is also a good strategy to increase aircrafft utilization by improving flight connection opportunities. When it comes to the application in the studies of schedule recovery, cruise speed control can be adopted to absorb the propagated delay or obtain the new aircraft swaps by adding additional ground time in the destination.

1.1.3 Robust aircraft maintenance routing problem (RAMRP) considering disruption risk during maintenance

In order to maintain the airworthiness of an aircraft, it should undergo regular aircraft maintenances before reaching the limit of usage parameters namely flight hours (FH), flight cycles (FC), or calendar days (DY). Regular maintenance tasks are categorized into four types of letter checks, i.e., A-check, B-check, C-check, and D-check, with varying scopes, frequencies, and durations. B-check is conducted every 300-600 flight hours; however, it is rarely mentioned in real-world ((Deng et al. 2020; Sriram and Haghani 2003). C- and D-check are scheduled at relatively large intervals of 18-24 months and 6-10 years, respectively, and require the aircraft to be out of service for a long period of time, for example one month. Accordingly, they are planned at the strategic decision level and not considered in airline operational scheduling problem (Al-Thani et al. 2016). On the other hand, A-check which is usually performed very 65 flight hours is the most frequent letter check. Generally, this interval varies according to the internal rules of airlines. A-checks involve visual inspection of major systems, such as landing gear, engines and control surfaces, and take six to eight hours (Al-Thani et al. 2016). Typically, A-check arrangements are tightly coupled with aircraft routes construction, which is known as aircraft maintenance routing problem (AMRP), due to their close relationship with the aircraft operational availability and daily scheduling (Cui et al., 2019).

Though, maintenance consideration dominates in AMRP, A-checks are assumed to be deterministic with identical maintenance tasks and durations for simplification due to the problem complexity. However, as a common practice, the A-check program is divided into several different packages, labeled in a sequential way (i.e., A1, A2, A3, ... for A-check) in order to increase the aircraft availability and the flexibility of grouping the maintenance tasks (Ackert 2010). Each package involves specific maintenance tasks and a different duration. Besides, variations in check duration are commonly seen and mainly induced by uncertainties encountered in maintenance A-checks, particularly from non-routine and unscheduled maintenance. According to the study of Hooplot and Ghobbar (2010), only a small percentage of the A-checks are completed within the scheduled time slot, which indicates a very low delivery performance of A-check. In this sense, the discrepancy between the planned duration and the actual duration becomes one of the biggest causes of disruption. As a typical case, the annual costs of flight delays directly caused by maintenance reached over $20 million dollars for a large airline (Cook et al. 2004). Moreover, it is found that the proportion of flight delays due to maintenance was up to 12.83%, and maintenance is one of the most crucial factors leading to long-lasting flight delays, by analyzing the flight data of European airlines from 2008 to 2014 (Zámková et al. 2017). Therefore, it is necessary and beneficial to integrate the consideration of maintenance disruption risk into the decision-making process in AMRP.

1.2 Research gap

1.2.1 Aircraft maintenance routing problem incorporating cruise speed control (AMRP-CSC)

From the operational side, we identify the research gaps as below:

1. In the current studies in AMRP, the flight flying times are assumed to be fixed during the route construction, which limits the flight connection opportunities and thus negatively affects the aircraft utilization improvement. In actual practice, shared by pilots, they may change the flying time by cruise speed control, such that they can mitigate a long-propagated delay or

reduce fuel consumption. Also, in many studies, the flight flying time of the schedule used as an input is not cast in stone and its variability is mainly achieved by cruise speed control (Duran et al. 2015; Gürkan et al. 2016; Şafak et al. 2017; Jalalian et al. 2019; Sun et al. 2020). They considered the impact of cruise speed control on the network connectivity of the flight's aircraft, crew, and passengers to downstream flights and applied it into schedule recovery problems.

2. Despite the positive impact of cruise speed control on aircraft utilization in aircraft re-routing problem, this similar impact in aircraft routing is seldom explored in the AMRP literature. Gürkan et al. (2016) have investigated the impact of cruise speed control on aircraft re-routing and showed that aircraft utilization can be increased by utilizing cruise speed control to improve flight connection opportunities. However, as the cruise speed control is adopted in the schedule re-optimization stage, it suffers from some limitations. First, in the re-optimization stage, the original schedule imposes significant restrictions on the generated schedule in order to preserve aircraft, crew and passenger connections, which limits cruise speed control from reaching its full potential in aircraft utilization. Furthermore, as the re-optimization stage is the final optimization stage prior to actual operation, it leaves relatively little flexibility in the utilization of the saved aircraft. Obviously, applying cruise speed control in aircraft routing can well overcome these limitations, which is studied in this work.

3. Regarding the methodology for AMRP, it is seen that matheuristics have seldom been studied in the AMRP literature. We know that the AMRP is very similar to the VRP since both of them can be reduced to TSPs. Recently, growing studies have applied matheuristics to VRPs and shown that matheuristics outperform the previous methods regarding solution quality and computation time (Villegas et al., 2013; Kramer et al., 2015; Brouer et

al.,2014; Pillac et al., 2013). Therefore, it is valuable and important to develop a matheuristic approach for AMRP-CSC to increase solution quality and computational efficiency.

To fulfill the research gaps, we propose a new AMRP model that incorporates cruise speed control to realize the flight flying time variability. Then, a matheuristis approach is designed for AMRP-CSC to improve solution quality and computational efficiency.

1.2.2 Robust aircraft maintenance routing problem (RAMRP) considering disruption risk during maintenance

From the robustness side, buffer time allocation and flight departure time retiming are common and useful robust approaches to mitigate the impact of uncertainty or disruption. However, all these studies focus on aircraft routing (Lan et al. 2006; Dunbar et al. 2012; Dunbar et al. 2014; Ahmed et al. 2017; Liang et al. 2015), while the maintenance regulations are either disregarded or used as constraints. Failure to capture the realistic consideration of maintenance checks, i.e., the heterogeneous maintenance tasks for A-check and check duration variability, results in a significant discrepancy between the planned duration and real duration, which could lead to serious disruption. Given the length of A-check, the protracted duration could cause an aircraft delay of up to a few hours, which will be fatal once the corresponding delay propagates to subsequent flights along the aircraft route. However, there is no literature available that addresses the RAMRP from the perspective of aircraft maintenance. On the other hand, the general trend in scheduling literature is placing more focus on including uncertainties in heavy maintenance, such as C- and D-checks, in the models (van der Weide et al. 2022). A-checks are closely related to aircraft routes, but C- and D-checks are typically conducted when an aircraft is out of service. Due to their stronger influence on aircraft operating availability than C- and D-checks, A-checks have a relatively significant impact on the development of airline operational schedules. In this regard, making a RAMRP model framework that takes the disruption risk from aircraft maintenance, which is examined in this study for the first time, into consideration is valuable and crucial.

1.3 Research objectives

This research mainly focuses on establishing a new AMRP model that incorporates cruise speed control, and subsequently on developing an effective method for handling real and large-scale problems. The research objectives can be summarized as follows:

1. To develop a AMRP model that incorporates cruise speed control. In the traditional model, the flight flying time is assumed to be fixed, which negatively impacts on flight connections during route construction. By considering flexible cruise times in routes construction, higher aircraft utilization and improved schedule flexibility can be achieved as a result of increased flight connection opportunities.

2. To develop a AMRP model that considers all the maintenance issues. These maintenance issues involve three maintenance requirements, as well as the workforce and working times of maintenance stations. In this context, the uncertainties of maintenance operations can be well captured, which ensures the smooth execution of maintenance operations. Thus, the schedule stability can be improved.

3. To develop an effective method for tackling real and large-scale problems. First, an IACO algorithm is designed with a new state transition mechanism that incorporates the node-based heuristic information, as well as a novel pheromone updating mechanism that provides the enhanced search efficiency and precision. Then, a matheuristic approach combing the IACO algorithm and exact methods is developed to further improve the solution.

4. To develop a new RAMRP that takes maintenance checks into account as the source of disruption and seeks to alleviate the impact of check duration variability on

aircraft maintenance routing. Specifically, an A-check program is divided into several task packages, each with varying maintenance tasks and durations. Then, the disruption risk caused by check duration variability is precisely quantified for each task package using a fuzzy risk assessment mode. A new robustness enhancement strategy is developed, of which the core idea is to identify the appropriate buffer time allocation for task packages based on the results of risk assessment. Besides, a robustness measurement, namely the total risk score, is proposed to construct the RAMRP model.

1.4 Main contribution and results

1.4.1 Aircraft maintenance routing problem incorporating cruise speed control (AMRP-CSC)

This research makes a significant contribution in terms of the operational reliability, the cost-effectiveness and the methodology, which are outlined as follows:

1. From the perspective of operational reliability, both schedule stability and flexibility are maintained simultaneously in our study. The explicit consideration of all aircraft maintenance issues captures the uncertainties of maintenance operations, hence ensuring schedule stability. Schedule flexibility has been achieved from two aspects. First, increased possibilities of aircraft swaps are obtained, which can create more re-routing alternatives in recovery. In addition, as our study reduces the required aircraft, the saved aircraft can be utilized in recovery when the number of used aircraft is different from the original schedule due to disturbances. In both situations, the schedule can be more easily repaired when facing disturbances, resulting in increased schedule flexibility.

2. From the perspective of cost-effectiveness, we can improve aircraft utilization and reduce the number of aircraft required. Due to more options in flight connections and cruise time compression of flights, the total idle time can be more decreased and the number of flights to be operated by one aircraft can be positively increased. This increase in aircraft utilization could reduce the total number of aircraft

necessary to fly the full range of flights. However, a crucial tradeoff between aircraft utilization and fuel--burn related costs must be examined. Computational studies are carried out to demonstrate the performance of the presented model. Compared with the traditional model, only a slight increase in the fuel--burn related costs is needed to yield a substantial improvement in aircraft utilization and thus reduce the number of required aircraft. We tested seven different networks on two aircraft types with significant differences in size and range and demonstrated that the approach is valid for all the test scenarios.

Moreover, the impact of compression level, compression interval, unit fuel cost and aircraft daily usage cost on the proposed model has been explored. The results show that, except for the compression level, the other three factors have no effect on the number of saved aircraft. This indicates that even when obtaining a new flight connection becomes much more expensive or an aircraft is much less valuable, the cost reduction brought by aircraft savings dominates the trade-off between aircraft savings and fuel--burn related costs all the time. This strongly supports the striking advantage of our model in which only a small amount of extra fuel consumption can achieve significant aircraft savings.

3. From the perspective of methodology, an IACO algorithm is designed first. Based on the traditional ACO, it makes improvements in following respects:

i. A new state transition mechanism incorporating the node-based heuristic information is employed. After the adoption of the flexible cruise time, it is necessary to provide the guidance for the selection of cruise times during the route construction. For the same flight leg, different cruise times result in different fight connection opportunities that directly affect the route construction. Therefore, the node-based heuristic information regarding the flight connection opportunities is considered in the new state transition mechanism.

ii. Three new strategies are applied to the pheromone updating mechanism. The first strategy, inspired by the study of Naimi and Taherinejad (2009), is used to decrease the pheromone increments from the beginning to the ending of a tour, since the freedom of flights selection is gradually restricted during the process of the route construction. Then, to better balance the trade-off between exploration and exploitation, the pheromone increments in each iteration are associated with current number of iterations performed, which is achieved by the second strategy. At last, to provide more precise directive information for subsequent searches, we distinguish the contribution of each route to the solution. Hence, the third strategy gives more pheromone increments on the arcs of the route covering more flight legs.

iii. Moreover, the pheromone structure and the heuristic function are modified to be more problem specific. Two different pheromone structures are set for maintenance arcs and non-maintenance arcs, respectively. Considering the positive impact of the flexible cruise time, we design a new heuristic function to give the priority to the feasible flight connections brought by cruise time compression.

Computational results reveal that the IACO algorithm can reach the optimality on small-size cases while achieving the upper bound obtained by the time constrained CPLEX solver on medium and large-size cases quickly. Moreover, the results verify that the IACO algorithm can obtain significantly better solutions in comparison with three promising meta-heuristic approaches.

Based on the IACO algorithm, a novel methodology, namely a matheuristic approach, is developed to solve the AMRP-CSC. It is composed of three components including an IACO algorithm, a SP procedure and a NS procedure. An IACO algorithm offers promising solutions for populating a pool of routes. Then, a SP procedure is adopted to form a superior solution by selecting the best recombination of the routes in the pool. A final improvement is achieved by a NS procedure which iteratively solves reduced AMRP-CSC instances. Each reduced AMRP-CSC instance is modeled as a Mix Integer Linear Programming (MILP) formulation and solved by the CPLEX solver. Two hybridization schemes, namely ACO-SP-NS-a and ACO-SP-NS-b, are designed based on these three components. ACO-SP-NS-a executes the three components in a sequential scheme. ACO-SP-NS-b, on the other hand, runs an integrated procedure that embeds the SP procedure inside the iterative process of the

IACO first, and then executes the NS procedure.

Computational experiments are conducted using the data extracted from the Bureau of Transportation Statistics (BTS). ACO-SP-NS-a and ACO-SP-NS-b are compared first, which turns out that a better compromise between solution quality and computation time can be obtained by applying ACO-SP-NS-a in small-size problems and ACO-SP-NS-b in medium and large-size problems. Later, further experiments are performed that compare this matheuristic with the CPLEX on small-size problems, and IACO, GA and SA on medium and large-size problems to verify its performance. Results reveal that it can quickly achieves the optimality when dealing with the small-size problems, while for the medium and large-size problems, it greatly surpasses the existing promising approaches regarding solution quality and computation time. Moreover, the largest size case that the proposed matheuristic can solve rapidly with the high-quality solution even contains up to 2100 flight legs.

1.4.2 Robust aircraft maintenance routing problem (RAMRP) considering disruption risk during maintenance

To our best knowledge, this is the first study that adopts different task packages rather than identical maintenance tasks for A-check in AMRP research. Based on this realistic consideration of A-check, a robustness enhancement strategy is provided to alleviate the disruption risk caused by variations in check duration. The core idea of this strategy is to give the task package with a higher risk level extra buffer time because each task package has a different risk level. A novel robustness measure that can precisely quantify the disruption risk is developed, which provides the insight on measuring schedule reliability and robustness from a risk analysis perspective. Using data from the Bureau of Transportation Statistics (BTS), we investigate the proposed model from two aspects, namely the restriction on aircraft utilization and the risk-averse attitude of airlines, both of which have a significant impact on the solution quality. The former limits the maximum allowable reduction in delay risks while the latter determines the willingness of the proposed model to pay more for a further mitigation of delay risks. Thus, our study demonstrates the advantage of the proposed model and highlights the importance of being properly risk-averse in decision making.

1.5 Structure of thesis

This thesis is organized as follows. Chapter 2 provides a brief literature review associated with airline schedule planning and cruise speed control. Chapter 3 presents a new AMRP model that incorporates the cruise speed control. The impact of flexible cruise time on solution improvement is evaluated by comparing the proposed model with the traditional model. Then, an IACO algorithm is proposed to effectively tackle AMRP-CSC model in Chapter 4. Chapter 5, based on the IACO algorithm presented in Chapter 4, introduces a novel solution methodology, i.e., a matheuristic approach, for AMRP-CSC. With a growing importance in the robustness of airline schedule, Chapter 6 propose a new RAMRP model that takes maintenance checks into account as the source of disruption and seeks to alleviate the impact of check duration variability on aircraft maintenance routing. Finally, Chapter 7 summarizes the conclusions drawn from the research findings. Additionally, it concludes with a discussion of the limitations of current research and the direction of future works.

Chapter 2 - Literature Review

Our study is mainly related to two streams of literature: the airline schedule planning problem and cruise speed control.

2.1 Airline schedule planning

As mentioned in chapter 1, the airline scheduling problem has been divided into four subproblems, due to its large-scale nature:

1. The flight scheduling problem (FSP) is to provide a list of flight legs, each characterized by a specified timetable, in order to meet the market demand.

2. The fleet assignment problem (FAP) is responsible for allocating aircraft types to individual flight legs. This stage attempts to match the aircraft type to the passenger demand for each flight leg in order to maximize the overall profit.

3. The aircraft maintenance routing problem (AMRP) aims to identify the sequence of flight legs for individual aircraft in a given fleet while adhering to the maintenance regulations mandated by the FAA.

4. The crew scheduling problem (CSP) is to deal with arranging crew members to each aircraft with the objective of minimizing the costs.

As illustrated in Figure 2.1, these four subproblems are handled sequentially, with the output of the former subproblem serving as input to the subsequent subproblems

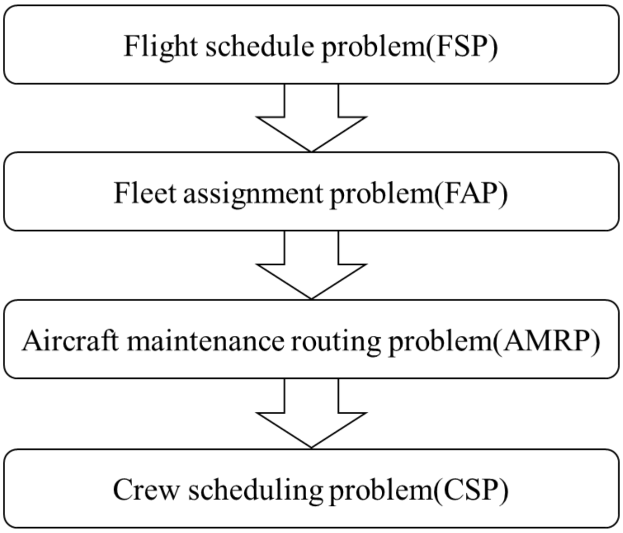

Flight schedule problem(FSP)
Fleet assignment problem(FAP)
Aircraft maintenance routing problem(AMRP)
Crew scheduling problem(CSP)

Figure 2.1: Airline planning process

In the subsequent sections, we will briefly review the literature in terms of four sub-problems and outline the characteristics of different models.

2.1.1 Flight scheduling problem (FSP)

The flight scheduling problem is the first problem to be solved and provides the basis for the subsequent problems. Typically, there are two independent stages in the flight scheduling process: timetable construction and timetable evaluation. These two phases are iterated until no more improvement can be achieved. After reviewing the studies regarding FSP, we divide these studies into two categories: the studies considering passenger demand and market share, and the studies considering robustness.

2.1.1.1 FSP considering passenger demand and market share

To create a profitable schedule, it is critical to consider passenger demand and market share when resolving the FSP. Yan and Young (1996) developed a framework to enable airlines to adjust their draft timetables and fleet routes in response to the fluctuating market demand conditions in the near future. In order to develop this framework, based on the basic multi-fleet model, three basic strategies were employed: removing multi-stop flight legs, modifying flight departure time and renting aircraft. A Lagrangian heuristic and a modified sub-gradient method are proposed to tackle the problems. The application of this framework in a major Taiwan airline shows it is efficient for actual operations. However, the market share consideration is ignored in this study. Yan and Tseng (2002) considered both the passenger demand and the market share, which are assumed to be the fixed. Their objective was to maximize the system profit.

Yan et al. (2007) designed a short-term flight scheduling model by deploying the time-space network technique and a passenger choice model, in which the variability in market shares was captured. Generally, the flight scheduling problem is formulated as an integer/mixed integer programming (IP/MIP) model. While they formulated the problem as a non-linear MIP model and then solve it by an iterative heuristic method. In each iteration, the market share of the target airline was modified and the flight scheduling problem with fixed market share was solved. Yan et al. (2008) considered both variability in market demand and market share by incorporating the passenger choice model into the stochastic-demand flight scheduling model (SDFSM). They developed the two heuristic algorithms that employ arc-based and route-base strategies to solve the model. However, only one-stop flights and single-fleet was taken into account. Jiang and Barnhart (2009) introduced a dynamic re-optimization model integrating flight retiming and re-fleeting. The flight schedule was redesigned at regular intervals, using information from revealed booking data and improved forecasts available at later re-optimizations. The results of computational experiments demonstrated that the significant potential profitability improvement was achieved.

2.1.1.2 Robust FSP

The purpose of a robust flight schedule is to reduce the susceptibility of the generated schedule to disruptions, which means that the irregularities that may happen in the schedule timetable can be absorbed. However, there are very few studies that focused on the robustness of FSP. Lee et al. (2007) improved the flight schedule robustness by re-timing. They formulated the problem as a multi-objective optimization problem and developed a multi-objective genetic algorithm (MOGA) for the problem. The generated schedule was evaluated using a simulation model SIMAIR 2.0 and the results indicated that the flight schedule, which are obtained by the application of MOGA, can achieve better operation costs and on-time performance. Burke et al. (2010) proposed a method for the multi-objective optimization of robustness objectives in airline schedules, in which the interaction between robustness objectives and their concurrent impact on the operational performance of the schedule were explored. They improved the multi-objective by retiming and rerouting simultaneously, while the fleet assignment was fixed. A multi-meme memetic algorithm was deployed to obtain diverse approximations of the Pareto optimal.

Sohoni et al. (2011) defined two service-level metrics, i.e., the flight service level and the network service level, for an airline schedule. Then, they developed a stochastic integer programming formulation, in which the objective was to maximize expected profit while maintaining the two service levels. A variant of this model was developed with the aim of maximizing the service levels and deriving the desired network profitability, and an efficient algorithm was proposed to solve these models. Extensive computational experiments demonstrated that a desired trade-off between service level and profitability was achieved by the proposed models and algorithms. Jiang and Barnhart (2013) designed a robust flight schedule where the number of potentially connecting itineraries was maximized. They modified an existing de-banked schedule, while satisfying two crucial conditions: (i) there is no peaking at a hub, and (ii) primary connecting itineraries are not disrupted. Two equivalent robust models with variable mathematical and computational properties were formulated. A decomposition-based approach employing a variable reduction technique and a variant of column generation was established to solve the problem.

2.1.2 Fleet assignment problem (FAP)

Given the flight timetables, the next step is the fleet assignment, which is closely related to revenue, along with operating costs, and plays an important role in achieving profitability for airlines. Referring to the past studies, we divide the studies into two parts: the basic FAP and its extension.

2.1.2.1 Basic FAP

Abara (1989) formulated the basic FAP as an integer linear programming (ILP) model, based on the connection network structure. However, they failed to deal with the problem for heterogeneous fleets due to the complexity of the problem. Rushmeier and Kontogiorgis (1997) solved the large-scale FAP. They modeled the problem as mixed integer multicommodity flow over networks where activities are encoded to link flight departures. The model successfully captured operational rules and allowed for the trade-off between the operational goals and revenue. Hane et al. (1995) presented a basic daily, domestic FAP and formulated the problem based on a time-space network structure. To reduce the complexity of the problem, preprocessing steps including node aggregation and isolated island, were developed. These studies provide the basis for the further improvement of the fleet assignment. In the next section, we will introduce the extension of basic FAP.

2.1.2.2 The extension of basic FAP

Berge and Hopperstad (1993) first proposed the Demand Driven Dispatch (D3) approach. They dynamically assigned aircraft of the same family to the flights so that the better match between the predicted final demands and fleet could be achieved. Desaulniers et al. (1997b) considered flexible departure time in the FAP and formulated the problem in two models, i.e., a set partitioning type formulation and a time constrained multicommodity network flow formulation. The linear relaxation of the former model was solved by a column generation approach while that of the latter model was tackled by a Dantzig-Wolfe decomposition approach. The upper bound was obtained by solving the linear relaxation of the former model. Different optimal branching strategies were proposed for tackling the problem. Rexing et al. (2000) improved flight connection opportunities by allowing variability in the flight departure time and thus achieved a more cost-effective fleet assignment. Two algorithmic approaches for solving the model were proposed. Good speed and simplicity was obtained by using the direct solution approach, whereas the memory usage was minimized by the iterative technique. Barnhart et al. (2002) presented a novel formulation and solution method for FAP, which captured the network effects. The benefit of the proposed approach was quantified in a case study.

Rosenberger et al. (2004) developed a relationship between short circles and hub connectivity. To achieve robustness, they reduced the hub connectivity and created many short circles. The results showed that both the planned operating cost and passenger spills could be minimized. Smith and Johnson (2006) imposed station purity on the FAP, which limited aircraft dispersion in the network. Therefore, robust solutions could be obtained. To increase the computational efficiency, station decomposition was proposed, and a primal-dual method was established to further enhance the performance of station decomposition. Due to the highly fractional characteristics of station decomposition solutions, a “fix-and-price” heuristic was introduced to identify the integer solution. Pilla et al. (2008) used a two-stage stochastic programming framework to model the FAP. The first stage assigned Crew-Compatible Aircraft (CCA) to flights. In the second stage, a probability distribution was employed to generate some equally likely scenarios while a deterministic FAP model was solved by using the current CCA for each scenario. Jacobs et al. (2008) incorporated origin and destination (O&D) network effects into FAP and proposed an O&D FAP approach. Both passenger flows over the network and demand uncertainty were addressed in this approach. Dumas et al. (2009) used the information from a passenger flow model to modify the objective function. They firstly generated fleet assignments and then used a modified passenger flow model to analyze them. This process was repeated to improve the objective function. Pilla et al. (2012) proposed multivariate adaptive regression splines cutting planes to resolve the FAP model. The computational time was significantly improved as compared to the L-shaped method due to the fast convergence.

2.1.3 Aircraft maintenance routing problem (AMRP)

The studies of the AMRP can be classified into three groups: tactical aircraft maintenance routing problem (TAMR), operational aircraft maintenance routing problem (OAMP) and robust aircraft maintenance routing problem (RAMP).

2.1.3.1 Tactical AMRP

The tactical side focuses on the generation of a specific rotation for each aircraft that will be repeated every day, while overlooking some maintenance restrictions. Kabbani and Patty (1992) addressed this topic for the first time and formulated the AMRP as a set-partitioning problem with side constraints. A sequence of flights is modeled by a connected flight string that satisfies the flow balance requirement, which originates and ends at a maintenance station so as to receive a maintenance check. Then a string-based model is developed where the decision variables represent the flight strings. However, the increased network size leads to exponentially growing number of strings, which makes it impossible to numerate all the strings and use one step solution approach to solve the model. Therefore, a two-step solution approach was proposed to resolve the problem. Gopalan and Talluri (1998), and Talluri (1998) used the lines of flying (LOF) concept that is described as a sequence of flights beginning and ending at airports in which the aircraft can receive maintenance overnight. A LOF-graph is constructed where the arcs represent LOFs, and a Euler tour represents a routing. A polynomial-time algorithm is developed for generating a Euler tour. They showed the problem becomes NP-complete if the LOFs are not fixed. In the study of Clarke et al. (1997), they formulated the AMRP as an asymmetric TSP and solved the problem using Lagrangian relaxation and subgradient optimization.

However, this approach didn’t guarantee finding an optimal solution. Liang et al. (2011) first define the daily AMRP by adopting the modified time-space network and propose a novel rotation tour time-space model. The model is highly compact and can be solved by the CPLEX solver due to its polynomial size.

In these studies, some of maintenance regulations were ignored. Moreover, it is difficult for airlines to repeat the routes due to fluctuating passenger demand. Therefore, operational AMRP appears, which will be introduced subsequently.

2.1.3.2 Operational AMRP(OAMRP)

The operational side, by contrast, aims to build a typical route accommodating the operational maintenance restrictions. Sriram and Haghani (2003) proposed the OAMRP by considering the maximum allowable number of consecutive flying days and the capacity of maintenance stations. They used Origin and Destination (O&D) pairs as an input to generate the route for each aircraft. To effectively tackle the OAMRP, a heuristic was developed. Furthermore, the model was extended by including the maintenance requirement, i.e., the maximum accumulated flying hours, but they failed to handle it due to the increased complexity. Sarac et al. (2006) deployed a branch-and-price method to solve the daily OAMRP while taking the resource availability of maintenance stations into account. Their aim was to minimize the total unutilized legal flying hours. However, they failed to tackle large-size problems because an exponential number of routes would be generated with the growth of the problem size. Haouari et al. (2013) presented a compact polynomial-sized model for the AMRP, while considering three maintenance requirements, including the maximum allowable flying hours, the maximum allowable number of take-offs, and the maximum allowable number of flying days. A reformulation-linearization technique was applied to linearize and lift the model formulation. Then, two root-node strategies were proposed to further augment the model formulation. The resulting formulations were polynomial in size and were efficiently solved by the commercial software. Başdere and Bilge (2014) formulated OAMRP as a multicommodity flow model. They distinguished the critical aircraft and the non-critical aircraft, thus the number of decision variables could be largely decreased in the model formulation. Their objective was to maximize the utilized flying hours of the critical aircraft. The branch-and-bound approach is adopted to solve small-scale problems and the compressed annealing approach is employed to tackle large-scale problems.

In more recent research, Al-Thani et al. (2016) make a modification that uses an aircraft node to present the aircraft based on the connection network of Başdere and Bilge (2014). Moreover, they simplified the problem complexity by introducing a graphical reduction procedure and valid inequalities. Thus, a more compact model with a polynomial number of variables and constraints is developed. The model is solved by a very large-scale neighbor-hood search algorithm in which a procedure for assessing the solution quality using a tight lower bound is proposed to increase its efficiency. Eltoukhy et al. (2017b) design an effective solution approach for the operational AMRP model that considers the maximum flying time and the maintenance resource availability. They validate this method by comparing it with the promising metaheuristic approaches including ACO, simulated sealing (SA), and genetic algorithm (GA). Based on this work, Eltoukhy et al. (2018a) extends the operational model by considering more maintenance constraints. They formulated the OAMRP as a mixed integer linear programming (MILP) model. The commercial software was adopted for small-scale problems and an effective heuristic was designed for large-scale problems. Cui et al. (2019) first included the objective of decreasing the number of required aircraft. They proposed an integer linear programming (ILP) model aiming to minimize the number of required aircraft and total unutilized remaining flying time. An improved heuristic, i.e., variable neighborhood search (VNS) algorithm, was adopted to solve the problem efficiently. However, the fixed flight flying times have a negative impact on flight connections during route construction, which poses limitations on this study. Bulbul and Kasimbeyli (2021) defined the big-cycle AMRP that generated a single route beginning and ending with the same flight leg. The problem was modeled as an asymmetric TSP over a connection network while considering the constraints of fleet size, maintenance violation and maintenance resource availability. A hybrid solution method was developed for the model that combined the Gasimov's modified subgradient algorithm and the ACO algorithm. In this approach, a formulation regarding the maintenance violation of the last iteration was added to the sharp augmented Lagrangia function which was subsequently minimized at this iteration by the ACO algorithm. This procedure would terminate until all the maintenance requirements were satisfied. Ruan et al. (2021) developed both a new ILP model and an innovative Markov Decision Process (MDP)-based model for the operational AMRP. The two models were solved by CPLEX solver and a learning-based algorithm, respectively.

2.1.3.3 Robust AMRP

The robust AMRP is to find a maintenance route that can be less susceptible to disruptions. Maher et al. (2014) focused on the aircraft route construction for a single day in order to mitigate the impact of schedule perturbations from the previous days. Then, single day AMRP (SDAMRP) was presented, in which a new modelling technique was used to count the number of maintenance misalignments in the solution. The model was further enhanced by using the recoverable robustness technique to increase the schedule recoverability. Ahmed et al. (2017) achieve the schedule robustness by inserting buffer times. They addressed the robust aircraft routing and retiming problem. Their objective was to obtain better airline on-time performance by generating the solutions that would be less susceptible to unpredictable disruptions. Finally, a novel two-level solution strategy was proposed for the problem. Jamili (2017) integrated aircraft routing with flight scheduling while assuming that the fleet assignment was determined. A robust mixed-integer mathematical model was developed, and then a heuristic algorithm based on Simulated Annealing (SA) is designed for the model. Eltoukhy et al. (2017a) studied OAMRP with flight delay consideration (OAMRPFD). To represent the non-propagated delay more explicitly, this study considered several potential scenarios rather than relying on the expected value. Hence a novel scenario-based stochastic framework for the OAMRPFD was developed. Although considering various scenarios greatly enlarged the problem size, the ACO algorithm could effectively solve the proposed model. Yan and Kung (2018) defined the flight delays within a predefined uncertainty set and intent to reduce the total maximum possible propagated delay. To solve the proposed model, exact and tractable solution methods were designed. Eltoukhy et al. (2019) incorporated the turn-around time reduction technique into AMRP to obtain the schedule robustness. Then, an ACO algorithm was proposed to solve the robust model.

2.1.4 Crew scheduling problem (CSP)

CSP is the last stage of the airline planning process. Due to tractability concerns, CSP is divided into two sub-problems: crew pairing problem (CPP) and crew rostering problem (CRP). The CPP generates a set of feasible pairings with the purpose of minimizing the total cost of crew assignment while adhering to various regulations. The CRP assigns the pairings generated in CPP to individual crew members in order to satisfy their skills, vacations, and other requirements. The literature regarding the CPP and CRP will be discussed as below.

2.1.4.1 Crew Pairing Problem (CPP)

A crew pairing is a tour consisting of a set of flight legs which originates and terminates at the same crew base. A pairing is separated by rest periods and a sequence of flight legs between two consecutive rest periods is defined as a duty. The crew pairings should cover all the flight legs, and one flight leg should be operated by exactly one crew pairing. To be legal, all the rules in the work convention and all the appropriate air traffic regulations should be satisfied. Generally, crew pairings are constructed over the planning horizon of several days, which can be severed as an input for the longer crew assignment problem.

Desaulniers et al. (1997a) modeled CPP as an integer, nonlinear multi-commodity network flow problem. The objective function was nonlinear and modeled without approximations. Based on an extension of the Dantzig-Wolfe decomposition principle, they developed a branch-and bound algorithm for solving the problem. Then, they performed the computational experiments utilizing the real-world Air France medium haul data by comparing with the expert system of Air France. Klabjan et al. (2001) which captured both cost and regularity by proposing a weekly crew scheduling model with regularity. Results demonstrated that significant improvement in solutions in terms of regularity and cost were achieved. Yan and Chang (2002) proposed a model and two scheduling networks, using real data from a Taiwan airline. Their objective was to minimize cockpit crew costs, by formulating a set-partitioning model to identify the optimal pairings. A column generation approach was designed to efficiently tackle the model. Shebalov and Klabjan (2006) increased the number of move-up crews, resulting in improved swapping opportunities for crew members and thus enhanced schedule robustness in crew schedule planning. A large-scale integer program was developed and solved by combining delayed column generation and Lagrangian relaxation. Computational results demonstrated the improved robustness of the generated schedule as compared to the traditional one. In order to further investigate the benefit of the robust schedule, they gave random disruptions to the schedule and concluded that robustness can result in reduced operational crew cost.

Deng and Lin (2011) formulated airline crew scheduling as a TSP model and solved the model by ACO algorithm. To validate the efficiency of the proposed algorithm, computational experiments were carried out using the real cases. Muter et al. (2013) defined robustness as the ability of accommodating additional flights by disrupting the original schedule as minimally as possible in real operation. In their study, several predefined recovery techniques were incorporated into the planned crew pairings to tackle the potential extra flights. They included the flight swaps between two existing pairings or the flight insertion. They formulated the problem as a conventional set covering problem and applied a row and column generation approach to tackle this difficult model.

More recently, Zeren and Özkol (2016) developed a novel column generation technique, a pricing network and a pairing elimination heuristic based on the previous studies. Then, they modeled the main problem as a set-covering problem and constructed a duty-flight overnight connection graph to model the pricing sub-problem as a shortest-path problem. Subsequently, the sub-problem was effectively resolved by combining the heuristic and exact algorithm. Deveci and Demirel (2018) solved the CPP with the goal of reducing the cost of crew pairings. A two-stage solution was proposed. The first stage generated all the possible legal crew pairings. Then, the best combination of crew pairings that has the minimal costs were derived by adopting an evolutionary algorithm. They proposed three different solution methods including two variants of genetic algorithm (GA) and a memetic algorithm (MA) that hybridizes GA with hill climbing. Their performance was compared, and the computational results showed that the presented MA offered the best solutions.

Antunes et al. (2019) captured the delay propagation via crew connections and the cost structure of the pay-and-credit crew salary scheme in detail. Then they found a better trade-off between the deterministic part of the planned costs and the expected costs of delay and disruption. A MILP model was developed and effectively solved by a novel solution method. Sun et al. (2020) used the heteroscedastic regression model to explore the characteristics of flight flying time. They incorporated the interdependent structures of the departure and arrival times of the consecutive flights into the CPP model. To deal with the model, a column generation-based algorithm was presented. Computational experiments using real data showed that the reliability of the crew pairings was sufficiently improved. Quesnel et al. (2020) proposed the CPP with complex features (CPPCF), which considered crew preferences in CPP. They identified six pairing features associated with crew preferences which were beneficial for the CRP, and the pairing with these features were rewarded by the objective function. A column generation algorithm was proposed to solve the CPPCF.

2.1.4.2 Crew rostering Problem (CRP)

The CRP determines a work schedule that consists of a sequence of duties for individual crew number while satisfying collective agreements and security regulations. As compared to the CPP, CRP receives less attention in the academic literature, and the offered models are extremely simplified.

Kohl and Karisch (2004) provided a thorough description of the real-world CRP and mathematical models that included various constraints and objectives, highlighting the complexity of real-world CRP . Moreover, the solution approaches that were utilized in the commercial crew rostering system were presented. Cappanera and Gallo (2004) modeled CRP as a multi-commodity flow problem. To reduce the graph dimensions, a preprocessing procedure was proposed. Furthermore, several families of valid inequalities that are computationally effective were introduced to tighten the linear programming formulation. Maenhout and Vanhoucke (2010) attempt to assign a personalized roster to individual crew member with the goal of decreasing the operational costs while satisfying the schedule impartiality and fairness. They developed a Dantzig–Wolfe decomposition-based formulation for CRP. The original problem formulation was broken up into a master rostering problem and a sub-problem. A scatter search algorithm was proposed to tackle the problem. All characteristics of the procedure were investigated by the detailed computational experiments. Moreover, comparisons with an exact branch-and-price algorithm and a steepest descent variable neighborhood search were made. Zhou et al. (2020) considered both fairness and satisfaction simultaneously in CRP. A multi-objective ACS (MOACS) was developed to solve the multi-objective CRP. At first, the fairness and satisfaction objectives were optimized by two ant colonies respectively. Then, in order to prevent ant colonies from focusing exclusively on their own objectives, a novel heuristic strategy that incorporates three different types of heuristic information schemes was developed. Finally, a local search strategy including two types of local search for fairness and satisfaction was proposed to find the global pareto front.

2.1.4.3 Integrated CPP and CRP

The airline crew scheduling problem is divided into two sub-problems, i.e., CPP and CRP, which are solved sequentially, considering its size and complexity. However, this may result in remarkably suboptimal solutions. Therefore, an increasing studies focus on the integrated problem.

Saddoune et al. (2012) integrated the CPP and the CRP with the goal of minimizing the total crew costs. They modeled the subproblems as shortest path problems with resource constraints over a complex network structure. Then, the integrated problem was solved by the method combining column generation method and dynamic constraint aggregation. Computational results showed that the proposed method could solve the small and medium-scale problems within reasonable computational times. Zeighami and Soumis (2019) first presented an extension of CPP, which considered the requests of pilot and copilot vacations. Then, they integrated the CPP with the CRP, which was solved by the method combing Benders' decomposition and column generation. The computational experiments were performed utilizing the real-world data from a major U.S. carrier.

2.1.5 Integrated airline schedule planning problem

As the interdependencies among these problems have been ignored by the sequential approach, and may result in suboptimal, or infeasible solutions, growing number of studies focus on the integration of these problems.

2.1.5.1 FSP integrated with FAP

Lohatepanont and Barnhart (2004) proposed integrated models and solution algorithms for FSP and FAP. They captured the relationship between the demand and supply in FSP. Two models, i.e., the integrated schedule design and fleet assignment model and the approximate schedule design and fleet assignment model, were proposed. Sherali et al. (2013) proposed a model that integrated the FSP and FAP that considered path-based demands, flexible flying time, schedule balance, demand recapture, together with optional legs, and multiple fare-classes. They conducted a polyhedral analysis was performed so that the model formulation could be tightened by adding valid inequalities. A solution method that adopted the Benders decomposition method was designed for the proposed model and its efficiency were demonstrated by computational experiments. Pita et al. (2013) developed mixed-integer linear optimization model integrating FSP and FAP while accounting for aircraft and passenger delay costs in decision making. Moreover, both airline competition, airline cooperation and airport congestion were incorporated into the model. Dong et al. (2016) developed two models for the integrated FSP and FAP. The first model considered discrete price choices in order to obtain a linear model. Then, this model was further extended by including the itinerary price elasticity, which was effectively resolved by a heuristic algorithm. The experimental results were carried out by simulating various market scenarios and the results proved that a significant profit improvement could be achieved by the proposed model. Kenan et al. (2018) considered the high uncertainty in fares and passenger demand and developed a two-stage stochastic programming model for the integrated FSP and FAP. The fleet family for individual flight leg was decided in the first stage, and then the fleet type for individual flight leg was determined under the consideration of demand and fare implementation in the second stage. The sample average approximation (SAA) algorithm was employed to tackle the problem. Its performance regarding solution quality and computational time was validated by computational experiments.

2.1.5.2 FAP integrated with AMRP

Barnhart et al. (1998) captured costs regarding aircraft connections and complicating constraints in the integrated FAP and AMRP model to achieve robustness. A string-based model was developed and solved by a branch-and-price algorithm. Moreover, the proposed problem was reduced to the AMRP with the additional constraints that maintained the same aircraft utilization by limiting the number of fleets to one. This AMRP could also be resolved by the extension of their model and solution method. Haouari et al. (2011) investigated the integrated problem appearing in TunisAir. They formulated two different models, i.e., an assignment-based model and a set partitioning model and designed two exact methods, i.e., Benders decomposition and a branch-and-price method, for the two models, respectively. The computational experiments showed that the branch-and-price algorithm outperformed the Benders decomposition method regarding solution quality and computation time. Liang and Chaovalitwongse (2013) developed a novel network-based MILP model for the weekly AMRP (WAMRP) based on a new weekly rotation-tour network. This formulation made the model size increase linearly with an increasing schedule size. A diving heuristic was able to resolve the proposed model effectively by the tight LP relaxation. Furthermore, an integrated model was proposed to tackle the WAMRP and weekly FAP simultaneously.

2.1.5.3 AMRP integrated with CPP

Mercier et al. (2005) determined aircraft routes and crew pairings with the goal of minimizing the cost. Furthermore, the model was enhanced by incorporating robustness, which was achieved by introducing penalties for the connections that may cause delays. Two Benders decomposition methods were investigated, one of which solved the AMRP as the master problem and the other of which solved the CPP as the master problem. Weide et al. (2010) solved two original problems, i.e., AMRP and CPP, iteratively to generate solutions with a low cost and a high degree of robustness. This approach was further extended by considering the AMRP along with two types of CPPs, one of which involves technical crews and the other one of which involves flight attendants. Díaz-Ramírez et al. (2014) considered airlines that had a single fleet, maintenance station, and crew base. They solved the AMRP and CPP sequentially and in an integrated manner. The computational results showed that small improvements were achieved by the integrated manner, but the improvement drastically increased with a growing fleet size or schedule complexity.

2.1.5.4 FAP integrated with CPP

Sandhu and Klabjan (2007) integrated the FAP with CPP, but the constraints regarding the aircraft maintenance were ignored. A Benders decomposition method and a method combining Lagrangian relaxation and column generation were developed for the proposed problem. Computational experiments were performed utilizing the real-world data and the results demonstrated the better average performance of Lagrangian relaxation based method. Gao et al. (2009) discussed the integrated problem aiming at generating the more profitable and robust solutions within reasonable computational times. To achieve this objective, they captured the impact of fleet assignment on crew planning and then presented the crew scheduling in a tractable manner in the integrated model. Moreover, they investigated the effect of station purity, i.e., fleet purity and crew base purity, on solution quality and computation time. A new approach was developed, in which the fleet assignment was integrated with crew connections and station purity was imposed. The computational results showed that the crew-planning cost was reduced, and robustness was improved by considering crew connections and station purity.

2.1.5.5 Integration of three different planning problems

Salazar-González (2014) integrated fleet-assignment, aircraft-routing, crew-pairing and crew-rostering. The integrated problem was tackled in two stages. First, the integration of the fleet-assignment, aircraft-routing and crew-pairing over the one-day planning horizon was solved. This combinatorial problem was modeled as a 2-depot vehicle routing problem where vehicles represent aircrafts, and the drivers represent crews. An integer programming model was developed and solved by a heuristic algorithm. Based on the solutions derived by the first step, the CRP was addressed in the second stage. The solution approaches were successfully implemented in real-world instances. Shao et al. (2017) developed an airline planning model which integrated the fleet assignment, aircraft routing, and crew pairing problems. The fleet assignment and aircraft routing were formulated over the node-arc network of which the size was polynomial while the CPP was modeled as set-partitioning problem. A Benders decomposition method and some acceleration techniques were incorporated to solve the problem. Utilizing the real-world data, the computational results showed the benefits of the proposed approach. Jamili (2017) integrated flight scheduling, aircraft routing, and the fleet assignment problem. A novel MIP model. Moreover, they considered the uncertainties regarding flying times in operation and defined a novel robust method. A heuristic algorithm on basis of the Simulated Annealing (SA) was developed. Its accuracy and efficiency were validated by the randomly generated cases.

2.2 Cruise speed control

An increasing number of studies have considered cruise speed control as an effective tool in airline schedule recovery. What we learn about the cruise speed control is mainly based on these studies. According to the recovery stages involved, we divide those studies into two categories: studies related to schedule re-optimization and studies related to schedule disruption recovery.

First, studies involving in schedule re-optimization stage were presented. In the studies of Duran et al. (2015) and Şafak et al. (2017), cruise speed control was only deployed to increase the schedules' robustness by maintaining a desirable passenger service level. Duran et al. (2015) studied the flight scheduling problem, and used cruise speed control to reduce flying time, thus additional ground time was added at the destination to cope with potential disruptions. Since there is a limitation of compression amount of cruise times, it worked simultaneously with idle time insertion. However, they did not consider the CO_2 emission cost. Şafak et al. (2017) studied the integration of fight scheduling and fleet assignment. To reduce additional fuel consumption they considered re-fleeting, but spill costs were incurred. Unlike the aforementioned studies, except for the role similar to idle time insertion to increase robustness, in the study of Gürkan et al. (2016) cruise speed control was utilized to improve flight connection opportunities aiming at aircraft utilization improvement when incorporated in integrated flight scheduling and fleet-path assignment. However, it suffers from certain limitations due to the application stage of cruise speed control. First, in the re-optimization stage, the original schedule imposes significant restrictions on the generated schedule in order to preserve aircraft, crew and passenger connections, which limits cruise speed control from reaching its full potential in aircraft utilization. Furthermore, as the re-optimization stage is the final optimization stage prior to actual operation, it leaves relatively little flexibility in the utilization of the saved aircraft. Jalalian et al. (2019) develop a green model that integrates flight scheduling, fleet assignment and gate assignment. Being different from the previous studies in which passenger service level was ensured by chance constraint, in this study, they designed some indicators for quantifying passenger service level and a new objective function was proposed to measure it. Cruise speed control was employed to balance two conflicting objectives: minimizing the CO_2 emission cost and the passenger service level cost.

As for studies related to the schedule disruption recovery stage, accounting for networkwide integrated effects, cruise speed control along with other recovery techniques were adopted into a recovery optimization model. Their purpose was for mitigating the effect of disruptions. The trade-off between fuel consumption and disturbances brought by disruptions was examined. Aktürk et al. (2014) proposed a flight rescheduling problem. Since fuel and CO_2 emission cost functions are non-linear, a conic program was introduced to reduce the computational difficulty. This approach was further utilized in the studies of Duran et al. (2015), Gürkan et al. (2016) and Şafak et al. (2017). Arıkan et al. (2016) integrated the aircraft and passenger recovery problem, however, they did not consider the CO2 emission cost brought about by cruise speed control. Marla et al. (2017) considered discrete cruise speed options. In this study, they stated that cruise speed control added flexibility in the schedule recovery by increasing the number of swap possibilities. Arıkan et al. (2017) constructed a new network structure based on the flow of each aircraft, crew member and passenger and included cruise speed decisions. Unlike the work of Marla, this study did not limit the cruise time decisions to the discrete variables due to their smaller network representation.

Despite the popular application of cruise speed control in schedule recovery stages, it is seldom discussed in the airline schedule planning stage. To my best knowledge, this is the first study which applies cruise speed control in schedule planning.

Chapter 3 - An Aircraft Maintenance Routing Problem incorporating Cruise Speed Control (AMRP-CSC) Model

3.1 Introduction

The AMRP is crucial in achieving airline's profitability as it directly impacts on aircraft utilization. However, to maximize aircraft utilization, the aircraft maintenance route leaves little buffer time for absorbing even minor delays, resulting in severe disruptions, and thus rising delay costs. Therefore, in addition to increasing aircraft utilization, it is also an absolute necessity to improve schedule stability and flexibility so that airlines can better manage the disruptions. It is known that allowing variability in flight flying time in aircraft re-routing can achieve improved flight connection opportunities, thus higher aircraft utilization and enhanced schedule flexibility. Nevertheless, the similar impact of flying time variability on aircraft routing is under explored. In this chapter, a new AMRP model that incorporates cruise speed control to realize the flying time variability is formulated. Moreover, to improve schedule stability, the proposed model considers all aircraft maintenance issues including three aircraft maintenance requirements, as well as the workforce capacity and working times of maintenance stations. By the proposed model, we will achieve two aims. The first aim is to maintain schedule stability and flexibility while the second aim is to improve aircraft utilization, hence reducing the number of required aircraft. To demonstrate the performance of the proposed model, computational studies are conducted to compare it with the conventional AMRP model.

The remaining parts of the chapter is arranged as follows. Section 3.2 gives a detailed description of AMRP-CSC. A mathematical formulation of AMRP-CSC is provided in section 3.3. To tackle the AMRP-CSC model, an ACO algorithm is developed in section 3.4. In section 3.5, computational studies are conducted to compare the AMRP-CSC model with the traditional model, so that the performance of AMRP-CSC model can be validated. Finally, this chapter is concluded in section 3.6.

3.2 Problem description

In our study, we intend to solve the aircraft routing problem over a four-day horizon for a given fleeted schedule while considering flexible cruise times and aircraft maintenance issues. The aircraft maintenance issues refer to three aircraft maintenance requirements, as well as the workforce capacity and working times of maintenance stations. Three maintenance requirements include the maximum allowed flying time, the maximum allowed take-offs and the maximum allowed number of successive flying days since the last A-check (Eltoukhy et al. 2018a).

In the proposed model, we identify the cruise time of each flight, the sequence of flight legs called the aircraft route for each aircraft and the maintenance decisions as to when and where the aircraft are properly maintained. These decisions are made with the objective of maximizing aircraft utilization while considering the fuel-burn related costs associated with cruise speed control. A critical tradeoff between aircraft utilization and fuel-burn related costs must be examined. For this purpose, quantifying aircraft utilization together with fuel-burn related costs is a necessity in the model. In consequence, the aircraft usage costs, the idle time costs and the fuel-burn related costs are considered simultaneously in making these decisions.

Typically, a flight can be separated into two phases: the cruise stage and the non-cruise stage, a latter of which incudes taxiing out, taking off, climbing, descending, final approaching, landing, and taxiing in. Thus, a flight time is comprised of cruise time and non-cruise time. Due to local traffic and safety concerns, non-cruise time cannot be excessively compressed and thereby is assumed to be a fixed parameter. Cruise time is included in the AMRP model as a decision variable that is directly related to fuel-burn related costs. To investigate the role of cruise time on fuel-burn related costs, the fuel flow model of cruise stage proposed by EUROCONTROL (2013) is employed. For a flight i flown by an aircraft type AC, given its cruise time to be CT_i, the fuel consumption in kg can be calculated as:

$$F_i^{AC}(CT_i) = ce_1^{i,AC}\,(1/CT_i) + ce_2^{i,AC}\,(1/(CT_i)^2) + ce_3^{i,AC}(CT_i)^3 + ce_4^{i,AC}(CT_i)^2 \tag{3.1}$$

where $ce_1^{i,AC}$, $ce_2^{i,AC}$, $ce_3^{i,AC}$, and $ce_4^{i,AC}$ correspond to fuel consumption coefficients that are available in BADA user manual (EUROCONTROL 2013). Referring to EUROCONTROL (2001), 3.15 kilograms of CO_2 can be emitted by one kilogram of fuel burnt. Consequently, fuel-burn related costs can be defined as:

$$CF_i^{AC} = \left(C_{fuel} + \varepsilon C_{co_2}\right)F_i^{AC}(CT_i) \tag{3.2}$$

Where C_{fuel} is the unit fuel price, C_{co_2} is the unit CO_2 emission cost and ε is the constant related to CO_2 emission.

3.3 Model formulation for AMRP-CSC

In this section, we modify the connection network proposed by Eltoukhy et al. (2018a), which is the most widely used network structure in the studies of AMRP. Based on the modified connection network, the problem is modeled as a multicommodity network flow model. In this part, we start by elaborating the modified connection network. Then the scope of proposed model and the notations are defined. Finally, we represent the detail formulation of the proposed model.

3.3.1 Modified connection network

The original connection network (Figure 3.1), proposed by Eltoukhy et al. (2018a) to describe OAMRP, includes two main elements: node sets and arc sets. The node sets consist of a flight leg node set, a maintenance station node set, and a dummy node set that includes the source node and the sink node. The arc sets contain: 1) a set of flight connection arcs, 2) a set of maintenance arcs, and 3) a set of auxiliary arcs. The flight connection arc (i, j) is deployed to maintain the connection between flight leg i and j, begin a route or terminate a route. The maintenance arc(i,m) is incorporated into the network to ensure a maintenance operation in a maintenance station m for an aircraft after operating a flight leg i. The auxiliary arc(m,j) is designed to resume operating the subsequent flight leg j after finishing a maintenance operation in a maintenance operation m.

From the description of the original connection network, we note that the node set cannot represent flight legs with controllable cruise time. To reflect the controllable cruise time, each node is assigned a cruise time window in which several copies with different cruise times are placed at a specified interval. For each node, only one copy can be covered. In this way, the model is allowed to select the cruise time for each flight. Figure 2 displays the modified connection network which is the extension of the original connection network shown in Figure1.

The width of cruise time window is defined by compression level which is the maximum amount of compression applied to scheduled cruise time for a flight, expressed as a percentage (Duran et al. 2015). For instance, if the compression level is set to be 10%, the cruise time window will be [90%, 100%]. This means that if the scheduled cruise time of a flight is 210 minutes, by cruise speed control, the cruise time is allowed to vary from 189 minutes to 210 minutes. Obviously, the compression level is a parameter much related to the solution quality. A higher compression level allows for more connections, but more additional fuel consumption will result simultaneously. Moreover, due to various physical constraints, for example, the capacity for fuel storage and cabin pressure, there is a limit on cruise speed, followed by an upper bound on compression level (Aktürk et al. 2014).

Another important parameter is the node copy interval. Changing this parameter will impact on fuel-burn related cost and consequently the model’s trade-off. In the model with a finer interval, to maintain the same flight connection, the amount of cruise time compression may be much less, leading to lower additional fuel-burn related costs. However, some unnecessary node copies may be brought, which causes an explosion in problem size.

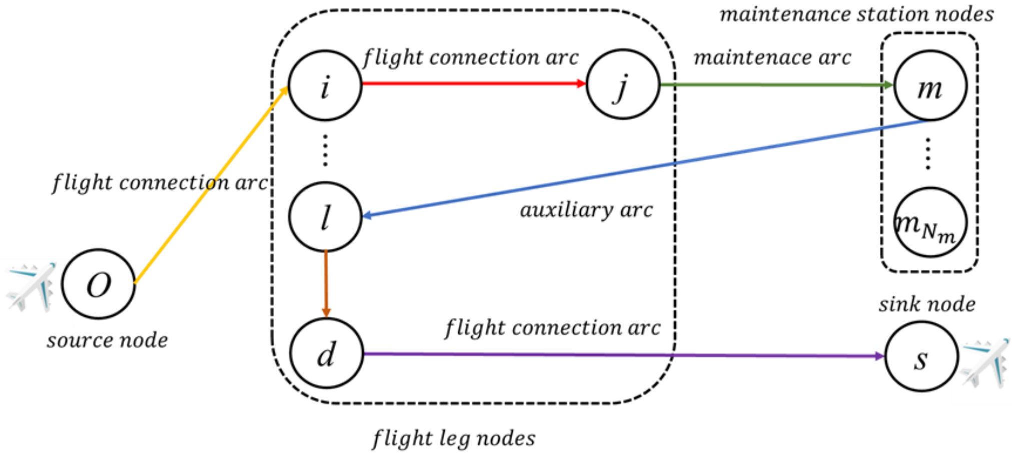


Figure 3.1: The original connection network

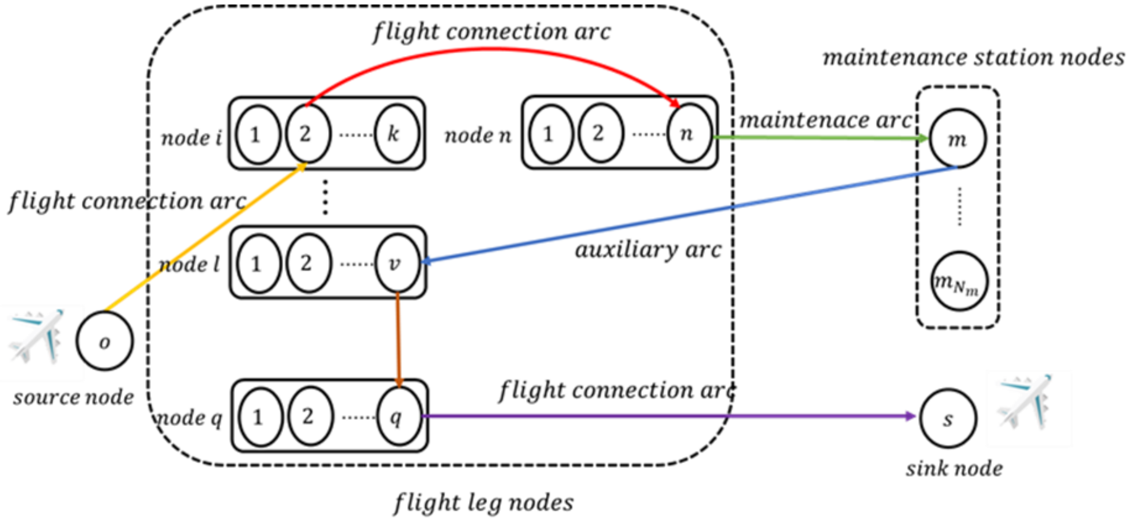


Figure 3.2: The modified connection network

3.3.2 Notations of the proposed model

The following notations are utilized in our model:

Sets

$i, j \in F$ Set of flight legs

$k \in K_i, n \in K_j$ Copy set of the flight leg that each copy represents one choice of cruise time duration under cruise

	speed control
$t \in T$	Set of aircraft of the same type AC
$m \in MT$	Set of Maintenance stations
$a \in A$	Set of airports
$v \in V$	Set of maintenance checks performed by each aircraft
o	Source node
s	Sink node

Parameters

DT_{ik}	Flight leg i copy k departure time.
FT_{ik}	Flight leg i copy k flying time.
AT_{ik}	Flight leg i copy k arrival time.
FC_{ik}^{AC}	Fuel-related costs when the aircraft of the type AC operates flight leg i copy k.
S_{ijt}^{kn}	Idle time between flight leg i copy k and flight leg j copy n for the aircraft t.
D_{ika}	$D_{ika} = 1$ if flight leg i copy k destination airport is the airport a and $D_{ika} = 0$ otherwise.
O_{ika}	$O_{ika} = 1$ if flight leg i copy k origin airport is the airport a and $O_{ika} = 0$ otherwise.
TA_{ijt}^{kn}	Turn-around time for the aircraft t connecting flight leg i copy k and flight leg j copy n.
F_{max}	Maximum allowed flying time since the last A-check.
TO_{max}	Maximum allowed take-offs since the last A-check.
Mb_{ma}	$Mb_{ma} = 1$ if the location of the maintenance station m is at the airport a and $Mb_{ma} = 0$ otherwise.
MAT	The duration of A-check.
M	A considerably large number.
WF_m	Workforce capacity of the maintenance station m.
ET_m	Closing time of the maintenance station m.

OT_m	Opening time of the maintenance station m.
C_{Daily}	Aircraft daily usage cost.
C_{Idle}	Unit idle time cost.
C_{fuel}	Unite fuel cost.
C_{co_2}	Unit CO_2 emission cost.
d	Planning horizon.

Decision variables

σ_{ik}	$\sigma_{ik} = 1$ if copy k for flight leg i is chosen and $\sigma_{ik} = 0$ otherwise.
$x^{n}_{ojtv} \in \{0,1\}$	$x^{n}_{ojtv} = 1$ if the aircraft t operates the flight leg j copy n as its first flight leg copy and $x^{n}_{ojtv} = 0$ otherwise.
$x^{kn}_{ijtv} \in \{0,1\}$	$x^{kn}_{ijtv} = 1$ if the aircraft t operates the flight leg i copy k and the flight leg j copy n consecutively before undergoing the v_{th} time maintenance check and $x^{kn}_{ijtv} = 0$ otherwise.
$y^{k}_{imtv} \in \{0,1\}$	$y^{k}_{imtv} = 1$ if the aircraft t operates the flight leg i copy k and undergoes the v_{th} time maintenance check consecutively and $y^{k}_{imtv} = 0$ otherwise.
$z^{n}_{mjtv} \in \{0,1\}$	$z^{n}_{mjtv} = 1$ if the aircraft t undergoes the v_{th} time maintenance check and operates the flight leg j copy n consecutively and $z^{n}_{mjtv} = 0$ otherwise.
$RTA_{mtv}>0$	The time when aircraft t finishes the v_{th} time maintenance check at the maintenance station m.

3.3.3 Model formulation

The new mathematical model can be characterized as follows based on its scope and notations:

$$\min \sum_{j\in F}\sum_{n\in K_j}\sum_{t\in T}\sum_{v\in V} C_{Daily} d\left(x_{ojtv}^{n}\right) + \sum_{i\in F}\sum_{k\in K_i} FC_{ik}^{AC}\sigma_{ik} + \sum_{i\in F\cup\{o\}}\sum_{k\in K_i}\sum_{j\in F\cup\{s\}}\sum_{n\in K_j}\sum_{t\in T}\sum_{v\in V} S_{ij}^{kn} C_{Idle} x_{ijtv}^{kn} \tag{3.3}$$

$$S_{ijt}^{kn} = DT_{jn} - AT_{ik} - TA_{ijt}^{kn} \quad \forall i,j \in F, \forall k \in K_i, \forall n \in K_j, \forall t \in T \tag{3.4}$$

$$S_{ojt}^{n} = DT_{jn} \quad \forall j \in F, \forall n \in K_j, \forall t \in T \tag{3.5}$$

$$S_{ist}^{k} = 24 \times 60 \times d - AT_{ik} \quad \forall i \in F, \forall k \in K_i, \forall t \in T \tag{3.6}$$

$$\textstyle\sum_{j\in F}\sum_{n\in N_j} x_{ojt1}^{n} \leq 1 \quad \forall t \in T \tag{3.7}$$

$$\textstyle\sum_{v\in V}\left(\sum_{j\in F}\sum_{n\in N_j} x_{jstv}^{n} + \sum_{m\in MT} z_{mstv}\right) \leq 1 \quad \forall t \in T \tag{3.8}$$

$$\textstyle\sum_{k\in K_i}\sigma_{ik} = 1 \quad \forall i \in F \tag{3.9}$$

$$\textstyle\sum_{t\in T}\sum_{v\in V}\sum_{k\in K_i}(\sum_{j\in F\cup\{s\}}\sum_{n\in N_j} x_{ijtv}^{kn} + \sum_{m\in MT} y_{imtv}^{k}) = 1 \quad \forall i \in F \tag{3.10}$$

$$\textstyle\sum_{j\in F\cup\{o\}}\sum_{n\in N_j} x_{jitv}^{nk} = \sum_{j\in F\cup\{s\}}\sum_{n\in N_j} x_{ijtv}^{kn} + \sum_{m\in MT} y_{imtv}^{k} \quad \forall i \in F, \forall k \in K_i, \forall t \in T, \forall v \in V \tag{3.11}$$

$$\textstyle\sum_{m\in MT} z_{mitv}^{k} = \sum_{j\in F\cup\{s\}}\sum_{n\in N_j} x_{ijtv+1}^{kn} \quad \forall i \in F, \forall k \in K_i, \forall t \in T, \forall v \in V \tag{3.12}$$

$$\textstyle\sum_{i\in F\cup\{o\}}\sum_{k\in K_i}\sum_{v\in V} y_{imtv}^{k} = \sum_{j\in F\cup\{s\}}\sum_{n\in N_j}\sum_{v\in V} z_{mjtv}^{n} \quad \forall t \in T\ ,\ \forall m \in MT \tag{3.13}$$

$$AT_{ik} + TA_{ijt}^{kn} - DT_{jn} \leq M(1 - x_{ijtv}^{kn}) \quad \forall i,j \in F, \forall k \in K_i, \forall n \in K_j, \forall t \in T, \forall v \in V \tag{3.14}$$

$$\textstyle\sum_{t\in T} x_{ijtv}^{kn} \leq \sum_{a\in A} D_{ika} O_{jna} \quad \forall i,j \in F, \forall n \in K_j, \forall k \in K_i, \forall v \in V \tag{3.15}$$

$$\textstyle\sum_{t\in T} y_{imtv}^{k} \leq \sum_{a\in A} D_{ika}\, Mb_{ma} \quad \forall i \in F, \forall k \in K_i, \forall m \in MT \tag{3.16}$$

$$OT_m - AT_{ik} \leq M(1 - y_{imtv}^{k}) \quad \forall i \in F, \forall k \in K_i, \forall t \in T, \forall m \in MT, \forall v \in V \tag{3.17}$$

$$AT_{ik} + MAT - ET_m \leq M(1 - y_{imtv}^{k}) \quad \forall i \in F, \forall k \in K_i, \forall t \in T, \forall m \in MT, \forall v \in V \tag{3.18}$$

$$RTA_{mtv} \geq \textstyle\sum_{i\in F\cup\{0\}}\sum_{k\in K_i}(AT_{ik} + MAT) y_{imtv}^{k} \quad \forall m \in MT, \forall t \in T, \forall v \in V \tag{3.19}$$

$$RTA_{mtv} - DT_{jn} \leq M(1 - z_{mjtv}^{n}) \quad \forall j \in F, \forall n \in K_j, \forall m \in MT, \forall t \in T, \forall v \in V \tag{3.20}$$

$$\sum_{t\in T} z_{mjtv}^{n} \le \sum_{a\in A} Mb_{ma}\, O_{jna} \quad \forall j \in F, \forall n \in K_j, \forall m \in MT, \forall v \in V \tag{3.21}$$

$$\sum_{j\in F}\sum_{n\in N_j}\sum_{m\in MT} FT_{jn} z_{mjtv}^{n} + \sum_{j\in F}\sum_{n\in N_j}\sum_{i\in F}\sum_{k\in K_i} FT_{jn} x_{ijtv+1}^{kn} \le F_{max} \quad \forall t \in T, \forall v \in V \tag{3.22}$$

$$\sum_{m\in MT}\sum_{j\in F}\sum_{n\in N_j} z_{mjtv}^{n} + \sum_{i\in F}\sum_{k\in K_i}\sum_{j\in F}\sum_{n\in N_j} x_{ijtv+1}^{kn} \le C_{max} \quad \forall t \in T, \forall v \in V \tag{3.23}$$

$$\sum_{i\in F}\sum_{k\in K_i}\sum_{m\in MT}\sum_{v\in V} y_{imtv}^{k} \ge 1 \quad \forall t \in T \tag{3.24}$$

$$\sum_{i\in F}\sum_{k\in K_i}\sum_{t\in T}\sum_{v\in V} y_{imtv}^{k} \le WF_m \quad \forall m \in MT \tag{3.25}$$

$$\sigma_{ik} \in \{0,1\} \quad \forall i \in F, \forall k \in K_i \tag{3.26}$$

$$x_{ijtv}^{kn} \in \{0,1\} \quad \forall i,j \in F, \forall k \in K_i, \forall n \in K_j, \forall t \in T, \forall v \in V \tag{3.27}$$

$$y_{imtv}^{k} \in \{0,1\} \quad \forall i \in F, \forall k \in K_i, \forall m \in MT, \forall t \in T, \forall v \in V \tag{3.28}$$

$$z_{mjtv}^{n} \in \{0,1\} \quad \forall j \in F, \forall n \in K_j, \forall m \in MT, \forall t \in T, \forall v \in V \tag{3.29}$$

$$RTA_{mtv} > 0 \quad \forall m \in MT, \forall t \in T, \forall v \in V \tag{3.30}$$

The objective function (3.3) is to minimize the sum of the aircraft usage costs, the fuel-burn related costs, and the idle time costs. Hereinto, the fuel-burn related costs are calculated by the equation (3.2) and the idle time is defined in constraint (3.4) -(3.6).

Constraints (3.7) -(3.10) define the constraints of flight coverage. Constraints (3.7) and (3.8) ensure the starting and ending of an aircraft route. Constraints (3.9) and (3.10) guarantee that exactly one copy can be selected by one flight leg and each flight leg must be covered by exactly one aircraft. Constraints (3.11) -(3.13) formulate the balance constraints, which maintains the balance when an aircraft visits a flight leg copy (constraints (3.11) and (3.12)) and a maintenance station (constrain (3.13)).

The time and place requirements for flight connections are presented in constraints (3.14) and (3.15). They ensure that two flight leg copies can be connected if and only if: 1) the origin airport of the former flight leg copy is identical to the destination station of the latter flight leg copy, 2) the time interval between the departure time of the former flight leg copy and the arrival time of the latter flight leg copy is not less than the minimum turnaround time. Constraints (3.16) -(3.18) guarantee that an aircraft can obtain a maintenance check after operating a flight leg copy if and only if: 1) the destination airport of the copy is a maintenance station, 2) the beginning time and completion time of the maintenance check must be within the working times in the maintenance station. Constraints (3.19) -(3.21) describe that after an aircraft finishes a maintenance check, the flight leg copy that it can perform consecutively must satisfy the requirements: 1) the origin airport of this flight leg copy is identical to the maintenance station, 2) the departure time of this flight leg copy must not be earlier than the completion time of A-check.

Constraints (3.22) -(3.24) specify the three maintenance requirements between the two consecutive A-checks including the maximum allowed flying time, the maximum allowed take-offs, and the maximum allowed flying days. Moreover, constraint (3.25) ensures that a maintenance station has sufficient workforce when an aircraft visit it. Finally, the domain of decision variables is indicated in constraints (3.26) -(3.30).

3.4 Solution method

The model becomes excessively large, particularly for realistically sized problems, due to the modeling of flexible cruise times. Therefore, a graphical reduction procedure that is helpful for reducing the size of network is designed first in section 3.4.1. Then, in section 3.4.2, an ACO algorithm is developed to solve the proposed model.

3.4.1 A graph reduction procedure

This is a strategy that can be used exclusively on networks that contain copies of specific nodes. A set of redundant node copies can be defined as node copies of the same flight leg that share the same flight connection opportunities. Among these node copies, the node copy that has the least amount of compression in cruise time dominates the other node copies, because an aircraft flowing across this node copy can make the same connection with the least fuel consumption.

Fig.3 shows a detailed portion of a flight network containing five flight legs (f_1, f_2, f_3, f_4, f_5) among which the flight leg f_1 has four copies (f_{11}, f_{12}, f_{13}, f_{14}). In this example, there are two sets of redundant copies: (f_{11}, f_{12}) and (f_{13}, f_{14}). Copies f_{11} and f_{13} can be deleted from the network without affecting the solution.

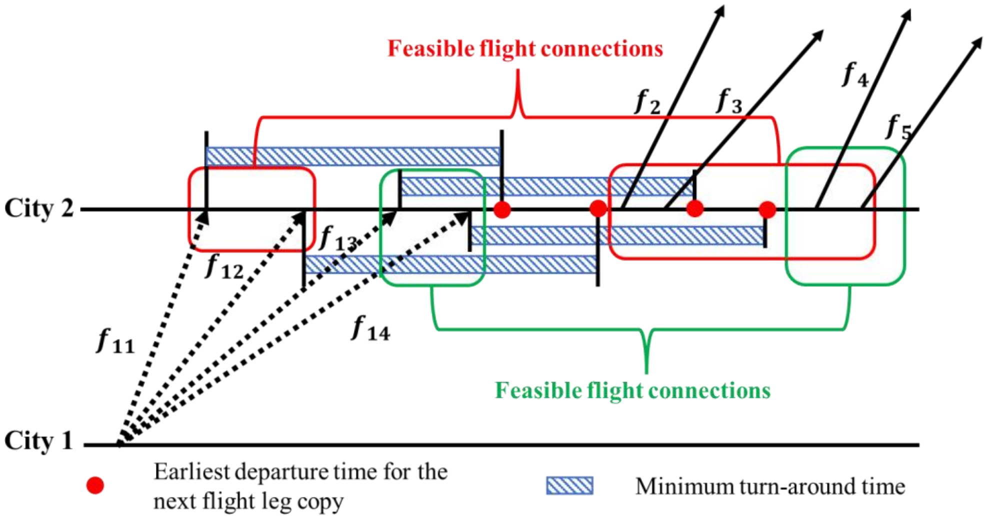

Feasible flight connections
f_2
f_3
f_4
f_5
City 2
f_{13}
f_{12}
f_{11}
f_{14}
Feasible flight connections
City 1
● Earliest departure time for the next flight leg copy
▨ Minimum turn-around time

Figure 3.3: A detailed portion of a flight network

3.4.2 An ant colony optimization algorithm

In this section, we develop an ACO algorithm to tackle AMRP incorporating cruise speed control. An ACO metaheuristic is chosen from two-fold considerations: 1) the problem size explodes due to the characteristic of flexible cruise times, while the ACO algorithm has been demonstrated to perform effectively for enormous and complicated network-based problems (Huang et al. 2018; Balseiro et al. 2011; Eltoukhy et al. 2019), 2) in the literature, the ACO algorithm has been successfully implemented for solving the robust AMRP, an extension of the basic AMRP, yielding high-quality solutions in short computational times (Eltoukhy et al. 2019).

The ACO meta-heuristic is inspired by the foraging behavior of ant colonies, which involves communicating via pheromones deposited along the way to discover the shortest route from a nest to a food resource. The pheromone information reflects the experience of previous ants in routes searching and changes dynamically to provide guidance for future ants. To employ an ACO algorithm to tackle the combinatorial optimization problems, we must model the problem as finding the shortest path on a weighted and constrained graph. Then, ants move over the graph, search for good paths using pheromones and problem-specific heuristic information, and deposit pheromones along the promising routes. Generally, the ACO algorithm is composed of two fundamental procedures: route construction and pheromone update. Next, we will detailly describe the ACO algorithm for AMRP incorporating cruise speed control based on the two procedures.

3.4.2.1 Route construction

In this procedure, an ant picks an aircraft ($t \in T$) and starts route construction at the source node. As it advances through the network, the state transition rule that is defined in equation (3.31) is utilized to select the next node. The selected flight leg copy is added to the route and all the copies of this flight leg are registered that will not be considered for future selection. At some point, there are no new flight leg copies to select due to various constraints, and the route is eventually ended at the sink node. Another new ant will be selected and construct its route by repeating the above process. This procedure continues until all the flight legs have been operated.

1. The state transition rule

$$f(x) = \begin{cases} arg\ \_max_{jn \in N_{ik}^{pq}} \left\{ \left[\tau_{ikjn}\right]^{\alpha} \left[\eta_{ikjn}\right]^{\beta} \right\}, if\ q \leq q_0 \\ p_{ikjn}^{pq}, \qquad if\ q > q_0 \end{cases} \tag{3.31}$$

$$\eta_{ikjn} = \frac{1}{\sqrt{S_{ijt}^{kn} + 1}} \tag{3.32}$$

In this equation, N_{ik}^{pq} represents the group of feasible nodes that can be covered after visiting the node ik. The term τ_{ikjn} and η_{ikjn} denotes the pheromones and the heuristic information on the flight connection arc connecting node ik and node jn, respectively. The value of η_{ikjn} is calculated using the equation (3.32), which indicates that the flight connections with smaller idle times are more attractive.

q is a random number with a uniform distribution in the range of 0 to 1, and q_0 is a predefined parameter. Their values determine the relative weight of exploration versus exploitation. The ant follows the most favorable path already constructed with a probability q_0 while performing the biased exploration based on the random proportional rule defined in equation (33) with a probability $(1 - q_0)$.

$$p_{ikjn}^{pq} = \left[\tau_{ikjn}\right]^{\alpha} \left[\eta_{ikjn}\right]^{\beta} \Big/ \sum_{dl \in N_{ik}^{pq}} [\tau_{ikdl}]^{\alpha} [\eta_{ikdl}]^{\beta} \qquad if\ jn \in N_{ik}^{pq} \tag{3.33}$$

Where the parameter α, and β control the influence of the τ_{ikjn}, the η_{ikjn} and the N_j^n, respectively.

3.4.2.2 Pheromone update

After each ant has completed its route construction, pheromones are updated using the following rule:

$$\tau_{ikjn,new} \leftarrow (1-\rho)\tau_{ikjn,old} + \varDelta\tau_{ikjn} \; i,j \in F, k \in K_i, n \in N_j \qquad (3.34)$$

Where ρ $(0 < \rho \leq 1)$ is a parameter representing the pheromone evaporation rate. The first term is utilized to achieve a uniform reduction of pheromones on all arcs, which helps an ant forget the bad routes previously taken. The second term is applied to increase the pheromones on the arcs that are only included in the currently best solution, aiming to direct the ants to select more promising elements during subsequent route constructions. It is calculated using the equation (3.35).

$$\varDelta\tau_{ikjn} = (Q \times R)/cost(R_{best}) \qquad (3.35)$$

The $cost(R_{best})$ denotes the best objective value obtained from the first iteration to the present iteration. Generally, the incremental pheromones on different arcs are the same in the conventional ACO. However, in the same solution, the routes that cover more flight legs are preferred and their arcs deserve more incremental pheromones. For this purpose, R is proposed in the term of incremental pheromones. It represents the number of flights covered by the route that includes arc $ikjn$. Parameter Q is adopted to adjust the incremental pheromones to control the balance between locally optimal convergence and randomization of the search.

3.5 Computational experiments

In this section, computational experiments are carried out to evaluate the performance of the proposed model. All the experiments were performed on a Windows 10 laptop with Intel Core i7, 2.6 GHz CPU, and 16-GB RAM. The implementations are coded in MATLAB R2016a. Section 3.5.1 describes the test instances and experimental setup. Section 3.5.2 demonstrates the superiority of the proposed model in terms of cost-effectiveness and operational reliability. Finally, section 3.5.3 presents some managerial insights that are derived based on the results analysis.

3.5.1 Test instances and experimental setup

In the experiments, seven flight schedules that contain 40, 83, 150, 222, 312, 410 and 613 flight legs are involved, which are generated using the data taken from the database of Bureau of Transportation Statistics (BTS 2020). Table 3.1 shows the overview of collected test cases.

To demonstrate the superiority of the AMRP incorporating cruise speed control model, we compare it with the traditional AMRP model. The traditional AMRP model is captured by two primary modifications of the presented model which is formulated in section 3.3.3. First, the constraint (3.9) regarding the cruise time selections is ignored. Second, only the scheduled cruise time is considered for each flight leg. To tackle this model, we modify the proposed ACO algorithm by neglecting the steps that describe the cruise time compression.

To study the performance of the proposed model on different aircraft types, two representative aircraft types, i.e., A320 111 and B767 300, are considered. Table 3.2 shows the fuel-burn related parameters for each aircraft type that are available in operational performance files from BADA (EUROCONTROL 2013). Moreover, the impact of four experimental factors, i.e., compression level, compression interval, unit fuel cost and aircraft daily usage cost, on the proposed model are investigated. Compression level is a crucial parameter due to its direct impact on the flight connection opportunities and consequently the route construction. The remaining three factors impact the model by acting on model's trade-off through changing the fuel-burn related costs or aircraft usage costs. Table 3.3 lists the settings for four experimental factors.

Before performing the experiments, the parameter settings regarding the ACO algorithm and the experimental setup are determined. First, we conduct a parameter tuning process for the ACO algorithm because its parameter settings significantly impact its performance. The parameters that greatly affect the performance of ACO algorithm are identified and then their levels are determined, which are listed in Table 3.4. Subsequently, the Taguchi method is utilized to identify the best parameter setting which are $\alpha = 1, \beta = 1, q_0 = 0.95, \rho = 0.15$ and number of ants =fleet size. Then, based on airline recommendations, the parameters regarding the experimental setup are set, which are provided in Table 3.5.

For each scenario, five replications are taken for the runs of the ACO algorithm. Considering a 2^k full-factorial experimental design, 80 random runs are generated on each test case

Table 3.1: Test cases

Test cases	Number of flight legs	Number of airports	Maintenance stations
1	40	14	7
2	83	24	10
3	150	42	16
4	222	53	18
5	312	56	18
6	410	57	21
7	613	57	21

Table 3.2: Aircraft parameters

Parameter	**Aircraft type**	
	B767 300	**A320 111**
Mass(kgs)	135000	62000
Reference wing area(m^2)	283.3	122.4
$C_{D0,CR}$	0.021	0.024
$C_{D2,CR}$	0.049	0.0375
Cf_1	0.763	0.94
Cf_2	1430	50000
Cf_{CR}	1.0347	1.095
MRC speed (km/h)	876.7	855.15
Base turn time (min)	40	28
Unit idle time cost ($)	147	136

Table 3.3: Experiment factors

Factors	Description	Levels	
		1(Low)	2(High)
A	Compression level	10%	15%
B	Compression interval	2.5%	5%
C	Unit fuel cost ($/kg)	0.33	1.26
D	Daily usage cost ($/day)	$C_{Idel} \times 60$	$C_{Idel} \times 600$

Table 3.4: Parameter settings for the ACO algorithm

Parameter		Level 1	Level 2	Level 3
Pheromone trail importance	α	1	2	3
Heuristic information importance	β	1	2	3
Exploration threshold	q_0	0.95	0.90	0.85
Evaporation rate	ρ	0.05	0.1	0.15
Ant size		0.5*Fleet size	Fleet size	1.5*Fleet size

Table 3.5: Parameter settings regarding the experimental setup

Parameter		value
Base turn-around time for the aircraft type AC	TA_{base}^{AC}	30 minutes
Maximum allowed flying time since the last A-check.	F_{max}	40 hours
Maximum allowed take-offs since the last A-check.	TO_{max}	8
The duration of A-check	MAT	6 hours
The scheduled non-cruise time for each flight		20 minutes

3.5.2 Advantages of the proposed model

In this section, the solutions obtained from AMRP incorporating cruise speed control model and AMRP model are compared. The results of aircraft savings, costs improvement, and percentage increase in fuel-burn related costs are illustrated in Figure 3.4, 3.5, 3.6, and 3.7. The costs improvement is calculated using the equation (3.36), while the calculation of percentage increase in fuel-burn related costs uses the negative of equation (3.36). To investigate the impact of experimental factor, a difference value that denotes the difference in costs improvement or percentage increase in fuel-burn related costs between two levels of each experimental factor is introduced. It is calculated by equation (3.37). Moreover, the comparison results between two levels of each experimental factor are displayed in Table 3.6, 3.7, 3.8, and 3.9. The Idle time (%) and Total (%) columns denote the improvement in idle time costs and total costs over the traditional model, respectively. The Fuel-burn (%) column represents the percentage increase in fuel-burn related costs over the traditional model. Finally, to investigate the impact of cruise speed control on flight connection opportunities and problem size, the number of remaining flight leg copies after the graph reduction procedure are provided in Table 3.10. According to the comparison results, the advantages of the proposed model will be analyzed from the perspective of cost effectiveness (achieved by aircraft savings and idle time costs improvement) and operational reliability (obtained by schedule stability and flexibility enhancement).

$$Costs\ \mathrm{i}mprovement = \frac{costs\ of\ traditional\ modle - costs\ of\ our\ model}{costs\ of\ traditional\ model} \tag{3.36}$$

$$difference\ value = \ result\ of\ level\ 2 - result\ of\ level\ 1 \tag{3.37}$$

3.5.2.1 The aircraft savings and costs improvement

On average, our model can achieve 5.3 aircraft savings, a 31.45% reduction in idle time costs and a 23.51% reduction in total costs with a 1.49% increase in fuel-burn related costs for B767 300, and 2.3 aircraft savings, a 23.79% reduction in idle time costs and a 19.31% reduction in total costs with a 0.84% increase in fuel-burn related costs for A320 111. The performance comparison between A320 111 and B767 300 is shown in Figure 3.8-3.11. Obviously, B767 300 has a better performance in aircraft savings and costs improvement but incurs more fuel-burn related costs, as compared to A320 111. This provides the insight that cruise speed control may have a greater impact on larger aircraft. Additionally, as shown in Table 3.10, the same compression level results in more remaining flight leg copies for B767 300 which further demonstrates this insight. This disparity is mainly brought by the difference in the base turnaround times of two aircraft types, i.e., A320 111 and B767 300. Next, we will analyze the performance of the proposed model in aircraft savings and costs improvement detailly.

Aircraft savings: From Figure 3.4, it is seen that the number of saved aircraft is not negatively affected by an increasing problem size. Moreover, only the compression level, out of the four experimental factors, has an impact on the number of saved aircraft, as can be clearly seen in Figure 3.4 (a). It is worth noting that the impact of wider compression interval and higher unit fuel cost can be exaggerated by the larger fuel burn rate of B767 300, but the number of saved aircraft is unaffected either, as shown in Figure 3.4 (b) (c).

The reason behind this behavior is twofold. On one hand, a higher compression level results in improved flight connection opportunities, as demonstrated in Table 3.10. In Table 3.10, it is observed that the number of remaining flight leg copies increases under a higher compression level except for case 1, case 3 for A320 111 and case 1 for B767 300. Accordingly, the number of saved aircraft remains the same in these cases. On the other hand, the degree of reduction in costs associated with aircraft savings is far more significant than the degree of increase in fuel-burn related cost. Therefore, the benefit of aircraft savings always dominates the trade-off of the proposed model even when obtaining a new flight connection gets more expensive or higher utilization of aircraft becomes less valuable. This is a good attribute for airlines as merely a small price is paid for reducing the number of required aircraft.

The idle time costs improvement: As can be found form Figure 3.5, the idle time costs improvement decreases when the problem size grows, which is resulted from the corresponding increase of cost base along with the growth of problem size (equation 3.36).

Regarding the impact of experimental factors, the two aircraft types, i.e., A320 111 and B767 300, are very similar. Except for the factor, i.e., daily usage cost, other three factors have a significant effect on the idle time cost improvement. As illustrated in Table 3.6, a higher compression level achieves increased idle time costs improvement due to the improved flight connection opportunities. In Table 3.7 and 3.8, we see that the performance in idle time costs savings is slightly improved with a finer compression interval or a lower unit fuel cost. This is because the model is encouraged to speed up the aircraft, allowing for more new connections to reduce idle times, when obtaining a new flight connection becomes cheaper.

The percentage increase in fuel-burn related costs: Despite the significant improvement in aircraft utilization, additional fuel-burn related costs will be incurred. As displayed in Figure 3.6, the fuel-burn related costs are slightly increased as compared to the traditional model, which is encouraging for airlines.

From Table 3.6, we observe that a higher compression level will result in an increased increase percentage of fuel-burn related costs. However, it is worthwhile to note that the average percentage increase in fuel-burn related costs drops from 0.94% to 0.85% for A320 111 in case 5. The underlying reason is that the improved flight connection opportunities enable more possibilities of route construction, resulting in a more efficient route with lower additional fuel-burn related costs. Then, the cost-effectiveness of improving flight connection opportunities in increasing aircraft utilization has been further demonstrated.

To achieve the same flight connection opportunity, the model with a finer compression interval may require less additional fuel consumption. Thus, the percentage increase in fuel-burn related costs significantly decreases in the model with a finer compression interval, as shown in Table 3.7. However, more additional fuel-burn related costs are incurred for B767 300 in case 4. This is because the model is encouraged to reduce the idle times by increasing flight connection opportunities when it is much cheaper to obtain new flight connections.

Despite the cost-effectiveness of the model with a finer compression interval, it should be noted that a finer compression interval may considerably increase the problem size. However, this is not the situation in our study. We can see from Table 3.10 that the number of remaining flight leg copies is the same with the two settings of compression interval, which indicates that the setting of a finer compression interval does not increase the network size after implementing the graph reduction procedure in our study. This provides a good reference to determine a cost-effective compression interval.

As previously stated, the model with a finer compression interval can result in a greater reduction in idle time costs with decreased additional fuel-burn related costs. In contrast to a finer compression interval, the model with a lower unit fuel cost increases the idle time costs reduction at a price of increased additional fuel-burn related costs, as illustrated in Table 3.8. The reason behind this behavior is that the proposed model tends to reduce idle times by compressing cruise times more aggressively when the unit fuel cost is lower.

The total costs improvement: As seen in Figure 3.7, the proposed model obtains significant total costs savings for two aircraft types under various scenarios due to its trade-off mechanism.

It can be interpreted form Table 3.6, 3.7 and 3.8 that the proposed model with a lager compression level, a finer compression interval or a lower unit fuel cost can achieve a better total costs improvement. This makes sense from two aspects that have been previously discussed. On one hand, the costs reduction achieved by increased aircraft savings or reduced idle times may significantly outweigh the increased additional fuel-burn related costs. On the other hand, more aircraft savings or better idle time costs improvement may be obtained even by less additional fuel-burn related costs. For the experimental factor of aircraft daily usage cost, it is interesting to find the model with a lower daily usage cost can yield a better total costs improvement, as presented in Table 3.9. This is owing to the lower corresponding cost base as the two settings of aircraft daily usage cost have no effect on the number of saved aircraft, the idle time costs improvement and the increase percentage of fuel-burn related costs.

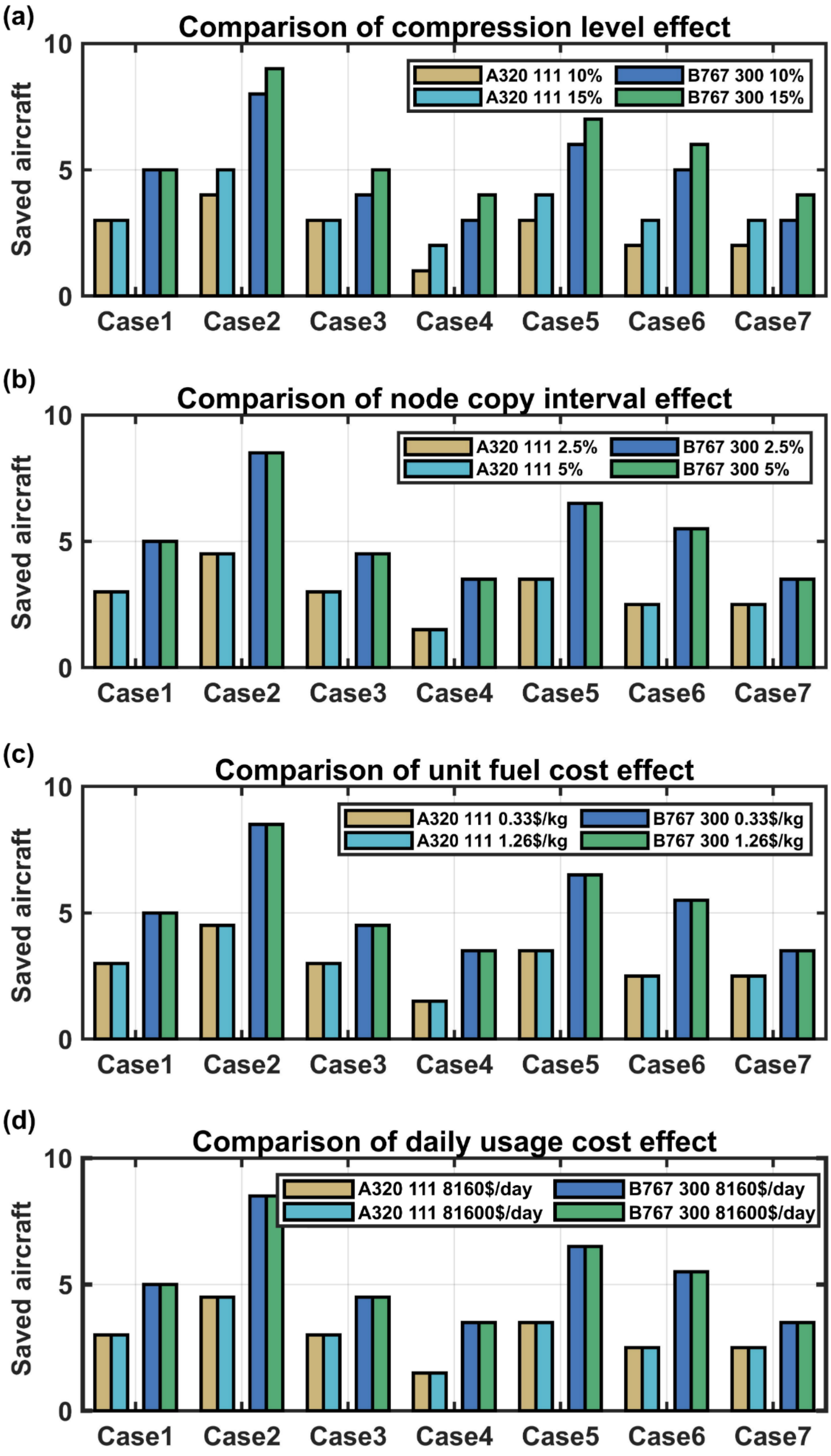


Figure 3.4: Aircraft savings

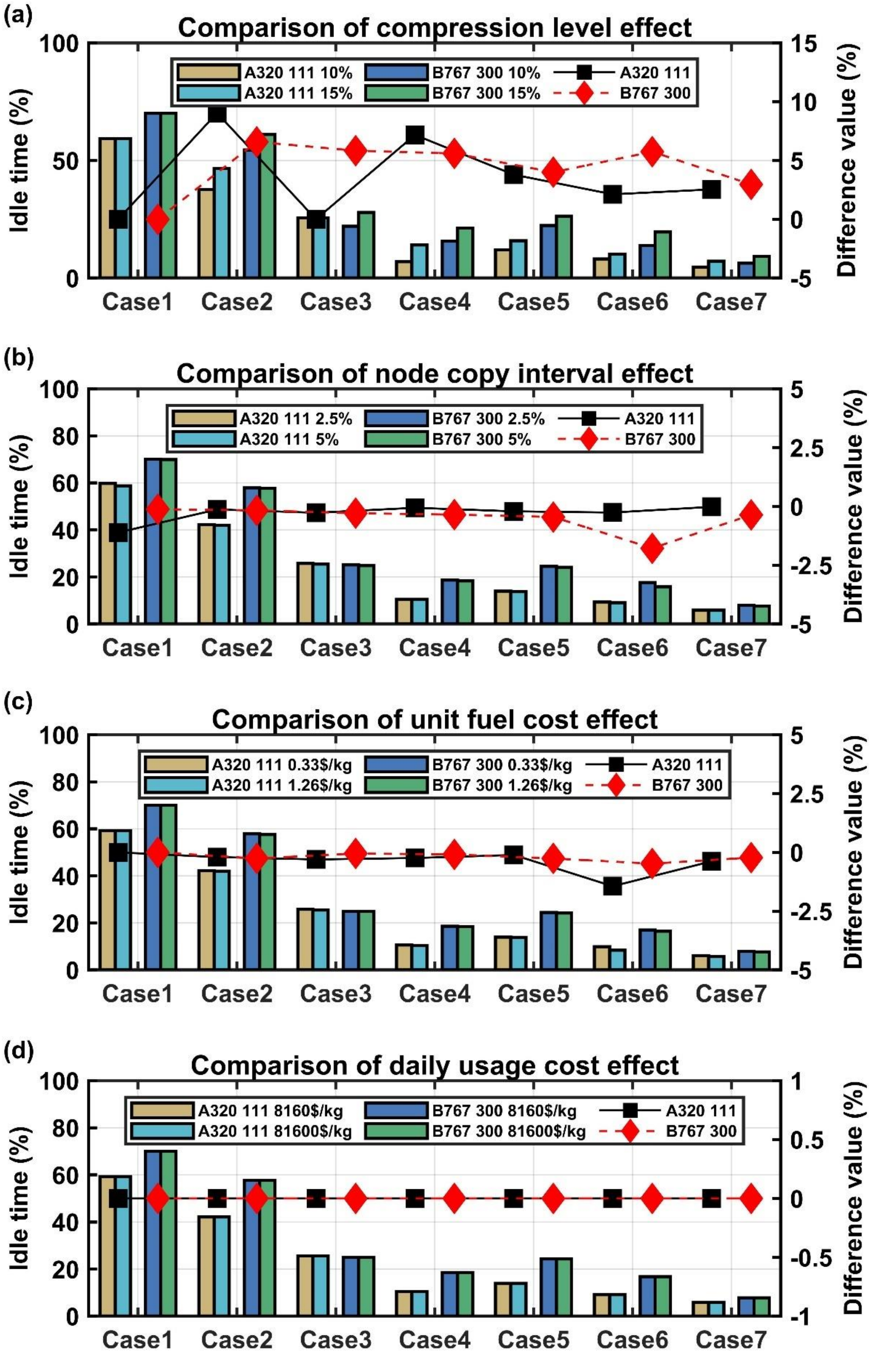


Figure 3.5: The idle time costs improvement

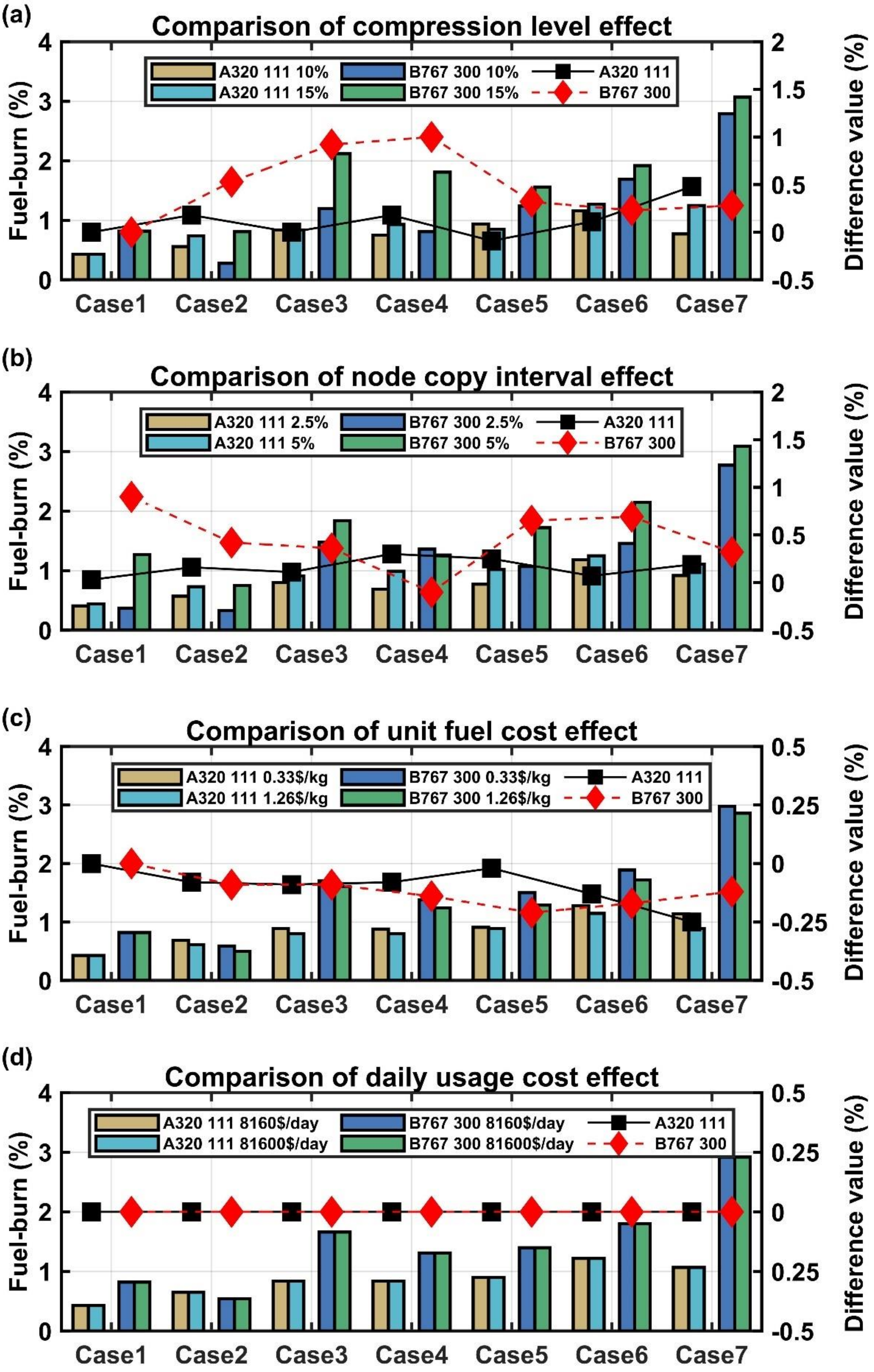


Figure 3.6: The percentage increase in fuel-burn related costs

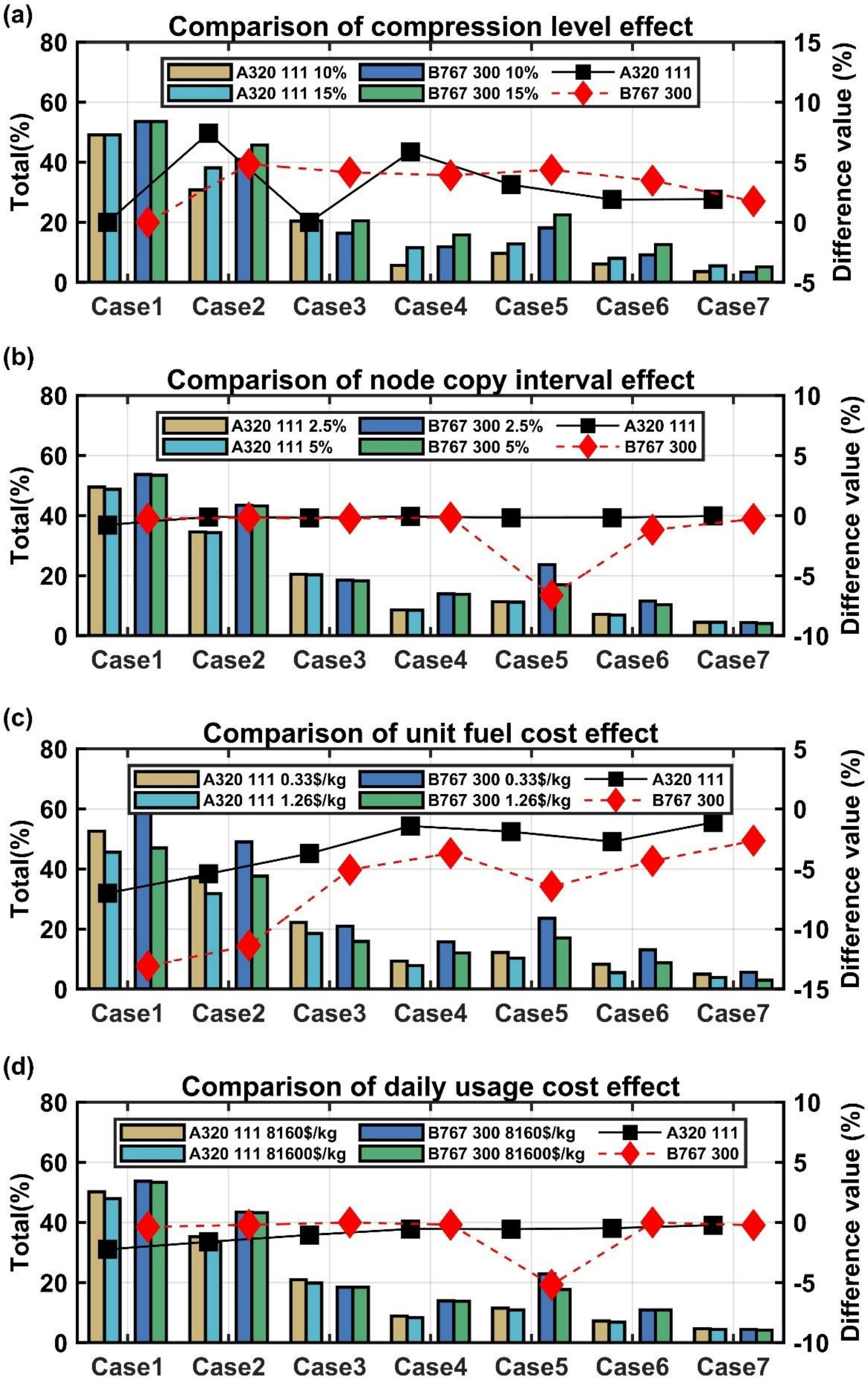


Figure 3.7: The total costs improvement

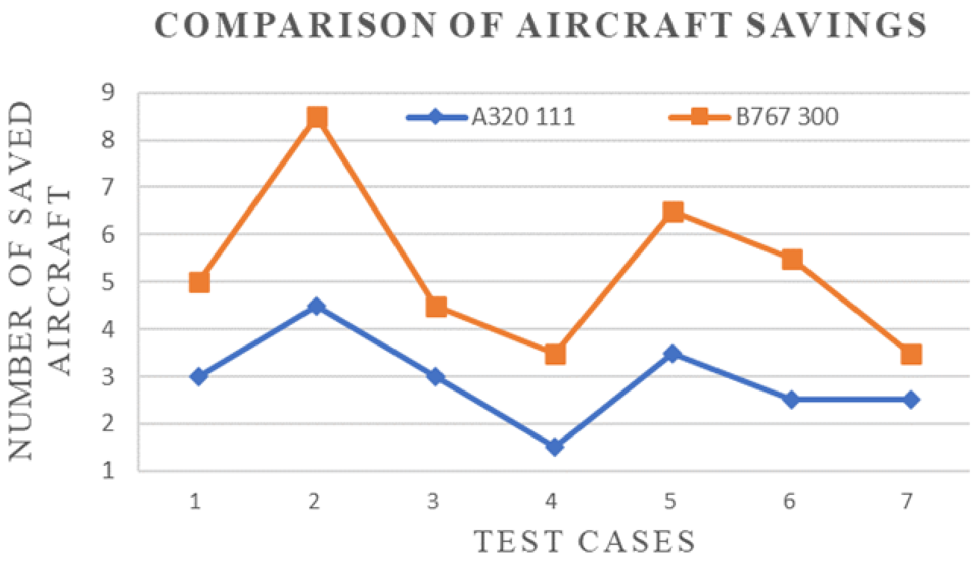


Figure 3.8: The comparison between A320 111 and B767 300 in aircraft savings

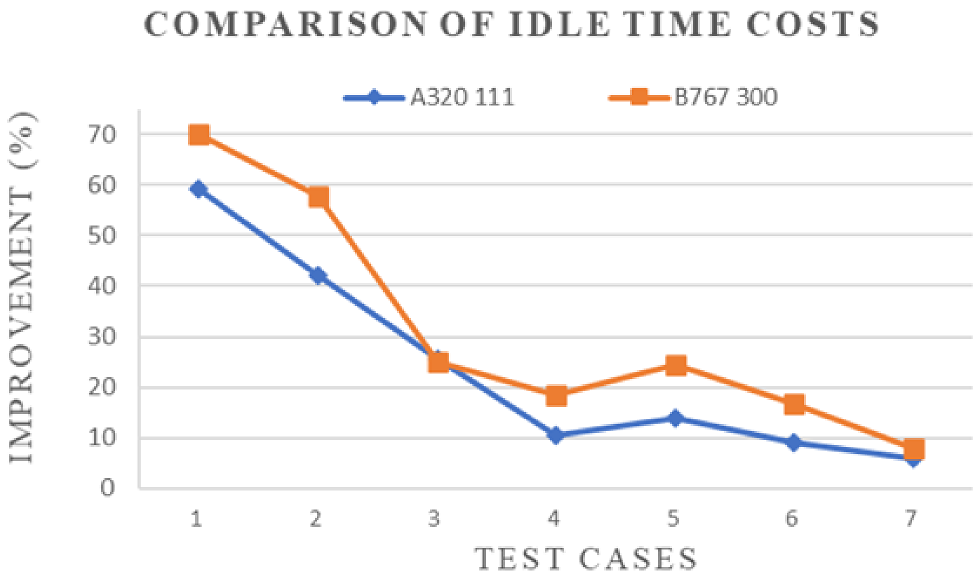


Figure 3.9: The comparison between A320 111 and B767 300 in idle time costs improvement

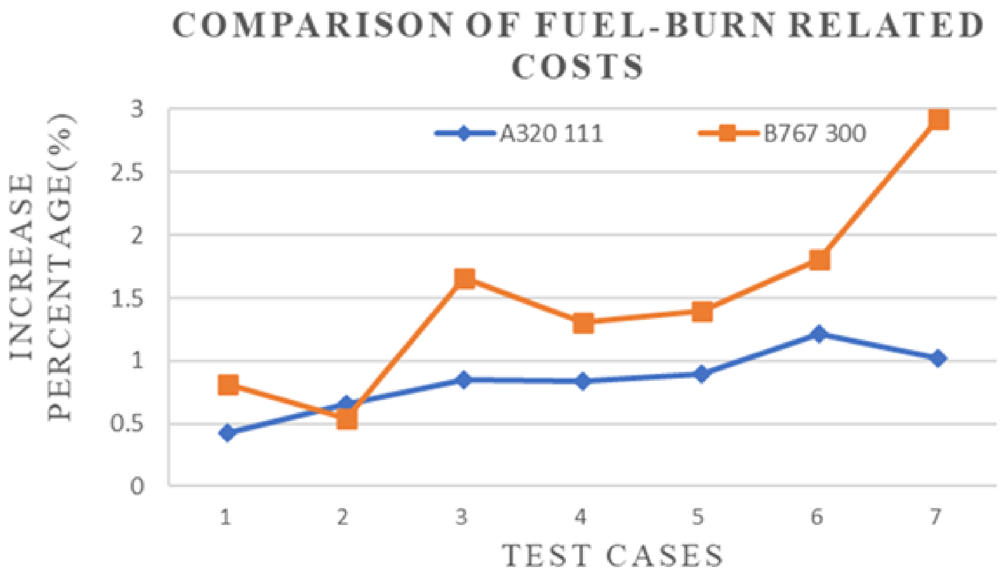


Figure 3.10: The comparison between A320 111 and B767 300 in percentage increase in fuel-burn related costs

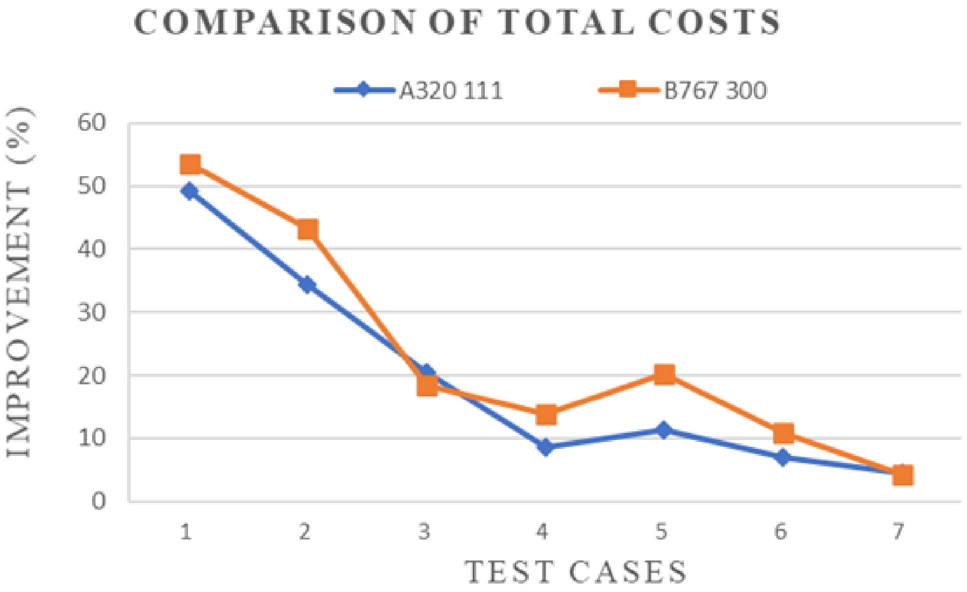


Figure 3.11: The comparison between A320 111 and B767 300 in total costs improvement

Table 3.6: Comparison of factor effects (Factor A)

		A320 111				B767 300			
Test cases	**Level**	**Saved aircraft**	**Fuel-burn (%)**	**Idle time (%)**	**Total (%)**	**Saved aircraft**	**Fuel-burn (%)**	**Idle time (%)**	**Total (%)**
1	1	3	0.43	59.27	49.1	5	0.82	70.01	53.54
	2	3	0.43	59.27	49.1	5	0.82	70.01	53.54
2	1	4	0.56	37.6	30.73	8	0.28	54.47	40.92
	2	5	0.74	46.63	38.14	9	0.81	61.05	45.76
3	1	3	0.84	25.6	20.38	4	1.2	22.06	16.33
	2	3	0.84	25.6	20.38	5	2.12	27.89	20.48
4	1	1	0.75	6.93	5.65	3	0.81	15.74	11.86
	2	2	0.93	14.1	11.5	4	1.81	21.33	15.77
5	1	3	0.94	12.01	9.67	6	1.24	22.33	18.12
	2	4	0.85	15.8	12.79	7	1.56	26.33	22.48
6	1	2	1.16	8.12	6.08	5	1.69	13.85	9.16
	2	3	1.27	10.24	7.97	6	1.92	19.59	12.63
7	1	2	0.77	4.65	3.53	3	2.79	6.3	3.37
	2	3	1.25	7.19	5.45	4	3.07	9.26	5.12

Table 3.7: Comparison of factor effects (Factor B)

		A320 111				B767 300			
Test cases	**Level**	**Saved aircraft**	**Fuel-burn (%)**	**Idle time (%)**	**Total (%)**	**Saved aircraft**	**Fuel-burn (%)**	**Idle time (%)**	**Total (%)**
1	1	3	0.41	59.83	49.49	5	0.37	70.07	53.67
	2	3	0.44	58.72	48.71	5	1.27	69.95	53.41
2	1	4.5	0.57	42.18	34.49	8.5	0.33	57.85	43.43
	2	4.5	0.73	42.05	34.38	8.5	0.75	57.68	43.25
3	1	3	0.8	25.74	20.48	4.5	1.48	25.12	18.53
	2	3	0.91	25.47	20.29	4.5	1.84	24.83	18.28
4	1	1.5	0.69	10.54	8.61	3.5	1.36	18.71	13.9
	2	1.5	0.99	10.48	8.53	3.5	1.26	18.36	13.74

5	1	3.5	0.77	14.01	11.32	6.5	1.07	24.56	23.62
	2	3.5	1.02	13.8	11.14	6.5	1.72	24.11	16.98
6	1	2.5	1.18	9.31	7.11	5.5	1.46	17.61	11.49
	2	2.5	1.25	9.05	6.94	5.5	2.15	15.83	10.3
7	1	2.5	0.92	5.93	4.51	3.5	2.77	7.96	4.39
	2	2.5	1.11	5.91	4.47	3.5	3.09	7.6	4.1

Table 3.8: Comparison of factor effects (Factor C)

		A320 111				**B767 300**			
Test cases	**Level**	**Saved aircraft**	**Fuel-burn (%)**	**Idle time (%)**	**Total (%)**	**Saved aircraft**	**Fuel-burn (%)**	**Idle time (%)**	**Total (%)**
1	1	3	0.43	59.27	52.61	5	0.82	70.01	60.08
	2	3	0.43	59.27	45.59	5	0.82	70.01	46.99
2	1	4.5	0.69	42.22	37.14	8.5	0.59	57.89	49.03
	2	4.5	0.61	42.02	31.73	8.5	0.5	57.63	37.65
3	1	3	0.89	25.75	22.24	4.5	1.7	25	20.94
	2	3	0.8	25.45	18.52	4.5	1.61	24.95	15.87
4	1	1.5	0.88	10.63	9.29	3.5	1.38	18.58	15.66
	2	1.5	0.8	10.39	7.86	3.5	1.24	18.49	11.97
5	1	3.5	0.91	13.96	12.18	6.5	1.5	24.46	23.53
	2	3.5	0.89	13.85	10.28	6.5	1.29	24.2	17.07
6	1	2.5	1.28	9.9	8.21	5.5	1.89	16.96	13.07
	2	2.5	1.15	8.46	5.48	5.5	1.72	16.48	8.72
7	1	2.5	1.14	6.11	5.01	3.5	2.98	7.9	5.58
	2	2.5	0.89	5.73	3.88	3.5	2.86	7.68	2.93

Table 3.9: Comparison of factor effects (Factor D)

		A320 111				**B767 300**			
Test cases	**Level**	**Saved aircraft**	**Fuel-burn (%)**	**Idle time (%)**	**Total (%)**	**Saved aircraft**	**Fuel-burn (%)**	**Idle time (%)**	**Total (%)**
1	1	3	0.43	59.27	50.22	5	0.82	70.02	53.35
	2	3	0.43	59.27	47.98	5	0.82	70.02	53.72
2	1	4.5	0.65	42.12	35.24	8.5	0.54	57.76	43.24
	2	4.5	0.65	42.12	33.63	8.5	0.54	57.76	43.44
3	1	3	0.84	25.6	20.9	4.5	1.66	24.98	18.4
	2	3	0.84	25.6	19.86	4.5	1.66	24.98	18.41
4	1	1.5	0.84	10.51	8.84	3.5	1.31	18.54	13.91
	2	1.5	0.84	10.51	8.3	3.5	1.31	18.54	13.73
5	1	3.5	0.9	13.91	11.51	6.5	1.4	24.33	17.71
	2	3.5	0.9	13.91	10.95	6.5	1.4	24.33	22.9
6	1	2.5	1.22	9.18	7.27	5.5	1.8	16.72	10.9
	2	2.5	1.22	9.18	6.79	5.5	1.8	16.72	10.89
7	1	2.5	1.01	5.92	4.6	3.5	2.92	7.79	4.14
	2	2.5	1.07	5.91	4.38	3.5	2.92	7.79	4.37

Table 3.10: Comparison of node copies for different compression levels and intervals

Test cases	Number of flight legs	A	B	A320 111	B767 300
1	40	1	1	46	44
		1	2	46	44
		2	1	46	44
		2	2	46	44
2	83	1	1	92	90
		1	2	92	90
		2	1	96	93
		2	2	96	93
3	150	1	1	163	177
		1	2	163	177
		2	1	163	189
		2	2	163	189
4	222	1	1	235	249
		1	2	235	249
		2	1	245	256
		2	2	245	256
5	312	1	1	331	366
		1	2	331	366
		2	1	338	385
		2	2	338	385
6	410	1	1	448	490
		1	2	448	490
		2	1	462	520
		2	2	462	520
7	613	1	1	683	754
		1	2	683	754
		2	1	708	803
		2	2	708	803

3.5.2.2 The schedule stability and flexibility enhancement

Despite of the cost-effectiveness of the proposed model, schedule stability and flexibility have also been investigated.

First, the model is solved by considering all the maintenance issues that involve three maintenance requirements, as well as the workforce and working times of maintenance stations. Violating any of three maintenance requirements will prevent the aircraft from operating flights, followed by delay or cancellation of subsequent flights in real operation. Moreover, ignoring the capacity of maintenance stations may result in an aircraft being held at a maintenance station due to the insufficient capacity. Similar situation may occur when an aircraft arrives at a maintenance station outside of the working times of this station. Both scenarios lead to severe aircraft delays in maintenance stations. In this context, the consideration of maintenance issues can well capture the uncertainties of maintenance operations, which ensures the smooth execution of maintenance operations. Thus, the schedule stability can be improved.

Second, the schedule flexibility can be gained in two ways. On one hand, we can see from Table 10 that the number of remaining flight leg copies increases in the proposed model as compared to the traditional model, which indicates the improved flight connection opportunities. For the same flight leg, if the copy with improved flight connection opportunities is selected, although it is utilized to increase the aircraft utilization, it provides the more possibilities of aircraft swaps and thus more re-routing options in the schedule recovery. On the other hand, it can be obviously observed that the required aircraft are significantly decreased. The saved aircraft can be utilized in recovery when the number of used aircraft is different from the original schedule due to the disturbances. As a result, in both of situations, the schedule can be easily repaired when facing the disruptions, and thereby the inherent schedule flexibility is built.

3.5.3 Managerial insights

From the results analysis, the application of AMRP incorporating cruise speed control provides some potentially managerial insights as well:

1. To achieve a more cost-effective solution, the decision on the proper compression level and node copy interval, which directly impact the fuel costs, is critical. In this vein, our study provides a good reference for airline decision makers through the investigation for the impact of the compression level and node copy interval on the proposed model.

2. In real operation, airlines are facing various uncertainties. Managing the disruptions caused by these uncertainties is significantly important for airlines to achieve operating cost reduction. Our study enables airlines to better manage the disruptions, and thus reduce the operating costs, by improving the schedule stability and flexibility.

3. Finally, our study is helpful for on-time performance (OTP) management. First, the achieved schedule stability and flexibility can positively promote the on-time performance (OTP) in real operation. Moreover, the OTP management can be enhanced by further balancing the trade-off between OTP and aircraft utilization. Note that the flight scheduling problem and the OAMRP are solved sequentially for the reason of tractability, which fails to capture the dependencies between two stages. However, the scheduled block time proposed in the flight scheduling stage strongly affects the aircraft route decision. For the excessive pursuit of OTP, airlines always propose a longer scheduled block time. This makes some connectable flights in practice become unconnectable in the aircraft routing model, leading to more ground time for an aircraft and consequently poor aircraft utilization. The aforementioned negative effect on OAMRP can be successfully mitigated in our model, as we shorten the scheduled block time within a reasonable range by cruise time compression. Therefore, the trade-off between OTP and aircraft utilization can be further balanced

3.6 Summary

This chapter investigates OAMRP incorporating cruise speed control. Due to the novel consideration of cruise speed control, we propose a new objective function for making a trade-off between aircraft utilization and fuel-burn related costs. Specifically, the optimization objective is to minimize the sum of aircraft usage costs, idle time costs and fuel-burn related costs. The flexible cruise time is modeled by permitting a flight’s cruise time to shift within the pre-specified cruise time window, which considerably enlarge the problem size. Thus, a graphical reduction procedure is a necessity. After the graphical reduction procedure, an ACO algorithm is designed to tackle the problem. Finally, we jointly identify the cruise time for each flight leg and the sequence of flight legs for each specific aircraft. The advantage of employing cruise speed control in the OAMRP model is examined by comparing the performance with the traditional OAMRP model in computational experiments.

The obtained results reveal that implementing our model is encouraging for airlines. First, the operational reliability has been enhanced in two aspects: schedule stability and flexibility. Capturing the maintenance uncertainties by considering maintenance issues achieves schedule stability. Moreover, the increased possibilities of aircraft swaps can result in additional re-routing alternatives in recovery. Saved aircraft can be utilized in recovery when facing disruptions. Both situations build flexibility into the schedule. On that base, from the perspective of cost-effectiveness, our model yields a substantial improvement in aircraft utilization and thus aircraft savings at the cost of a small increase in fuel consumption as compared to the traditional model. Furthermore, from the perspective of airline management, implementing our model is beneficial for both airline disruption management and OTP management. Finally, what matters most is that the attributes of our model have been verified to be appropriate for different aircraft types over different flight schedules.

Beyond all the above, our model has high potential with the rapid technological advancements in airline industry. Marais and Waitz (2009) stated that the fuel consumption per passenger kilometer has decreased by approximately 70% over the past four decades, indicating the potential for the further decrease in fuel consumption and CO_2 emissions with the improved engine fuel efficiency in the future. In this case, less additional fuel consumption is required to achieve the same cruise time compression, making it much cheaper to speed up an aircraft, and, as a result, implementing our model becomes more cost-effective.

Chapter 4 - An Improved Ant Colony Optimization (IACO) Algorithm for Aircraft Maintenance Routing Problem incorporating Cruise Speed Control (AMRP-CSC)

4.1 Introduction

For solving the AMRP, adopting the heuristic and metaheuristic approaches is more appreciated due to its NP-hard nature (Avci and Topaloglu 2016; Lin et al. 2021; Phoemphon et al. 2021; Chambari et al. 2021). Among these approaches, ACO is particularly appropriate for the AMRP in terms of two considerations. On the one hand, if considering the source nodes and the sink nodes on the connection network as the nests and the food resources, respectively, searching for optimal aircraft routes is quite similar to the foraging behavior of ants. On the other hand, AMRP is a heavily constrained problem. As ACO is a constructive metaheuristic, all constraints can be easily satisfied during the solution construction phase, making coding of ACO for AMRP much more straightforward. As a result, ACO algorithms have been widely and successfully applied in various variants of AMRP (Bulbul and Kasimbeyli 2021; Eltoukhy et al. 2017a; Eltoukhy et al. 2019; Eltoukhy et al. 2018b). However, despite the efficiency of ACO, applying the traditional ACO in AMRP incorporating cruise speed control is insufficient and unwise. The traditional ACO selects the next covered flight leg only utilizing the attractiveness of flight connections, while ignoring the information regarding the individual flight leg (i.e., flexible cruise times), which fails to optimize the cruise times. Moreover, the performance of ACO decreases dramatically in tackling large-scale problems, particularly when modeling flexible cruise time causes an explosion in problem size. Based on these observations, to tackle the problem more efficiently, an improved ant colony optimization (IACO) algorithm is developed. The CPLEX solver and three promising mate-heuristic approaches including the conventional ACO, simulated annealing (SA) and genetic algorithm (GA) are adopted as a comparison for studying the performance of the proposed algorithm.

The remaining part of this chapter is structured as follows. The model of AMRP-CSC is formulated in section 4.2. Based on the ACO algorithm in section 3.4, section 4.3 presents an IACO algorithm. To verify the performance of IACO algorithm, computational experiments are conducted in section 4.4. Finally, the section 4.5 summarizes this chapter.

4.2 Model formulation for AMRP-CSC

4.2.1 Modified connection network

To develop the mathematic model, we shall use the modified connection network shown in Figure 1, which is constructed as follows.

The node set consists of:

1. $\{o, s\}$ where o are the dummy source nodes and s are the dummy sink nodes.
2. $\{i1, \ldots, ik, \ldots, j1, \ldots, jn, \ldots\}$ where node ik represents flight leg i copy k. To model the flexible cruise time, for each flight leg i, several copies, of which each copy represents a choice of cruise time, are placed at a specified interval within a predefined cruise time window. During the route construction, only one copy can be chosen for one flight leg. For instance, the original cruise time is assumed to be 240 minutes for a flight leg i. The cruise time window and the interval are set to be $[192, 240]$ and 12 minutes, respectively. Therefore, there will be five copies for this flight leg which represent five choices of the cruise time including 192, 204, 216, 228 and 240, and only one copy can be covered in the route construction.

Regarding the arc set, it includes:

1. Non-maintenance arcs (ik, jn) which are designed to begin a route, connect two copies of flight legs successively or terminate a route.

2. Maintenance arcs (jn, lv) which are used to schedule a maintenance operation between two successive copies of flight legs.

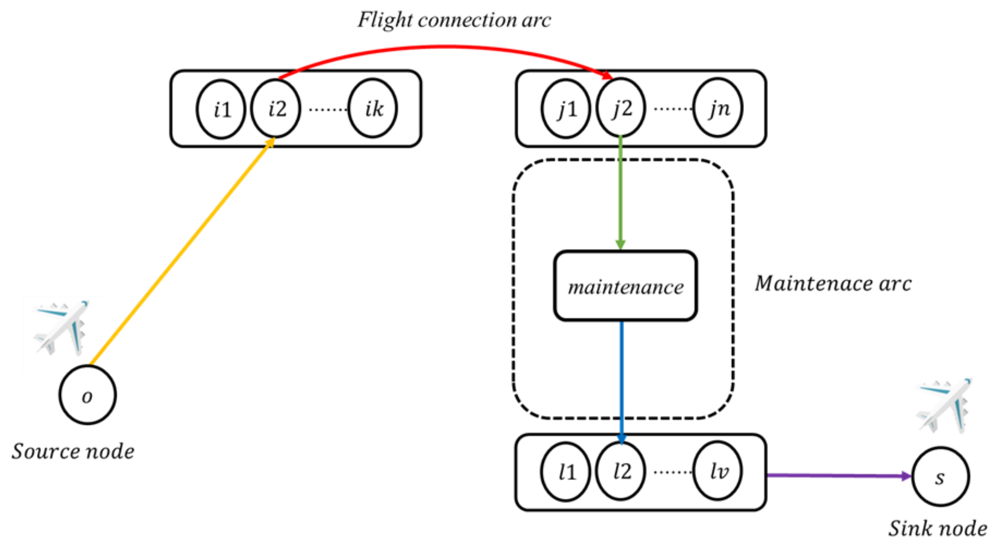


Figure 4.1: The modified connection network

4.2.2 Scope of the model and notations

Before building the mathematic model, we define the following scope:

1. A four-day planning horizon is adopted.
2. The A-check is considered in AMRP as it is performed more frequently.
3. The locations of maintenance stations are assumed to be the hub airports.

Then the notations used in our model are described below:

Sets

$i, j \in L$	Set of flight legs
$k \in K_i, n \in N_j$	Set of copies for each flight leg that one copy corresponds to one choice of cruise time
$t \in T$	Set of aircraft
$m \in M$	Set of maintenance stations
$c \in \{1 \dots \dots C\}$	Number of maintenance checks
$a \in A$	Set of airports
$o \in O$	Dummy source node
$s \in S$	Dummy sink node

Parameters

DT_{ik}	Departure time of flight leg i copy k.
FT_{ik}	Flying time of flight leg i copy k.
crs_{ik}	Cruise time of flight leg i copy k.
AT_{ik}	Arrival time of flight leg i copy k.
DS_{ika}	DS_{ika}=1 for flight leg i copy k departing from airport a and otherwise DS_{ika}=0.
AS_{ika}	$AS_{ika} = 1$ for flight leg i copy k arriving at airport a and otherwise $AS_{ika} = 0$.
TA_{ijt}^{kn}	Turnaround time for connecting flight leg i copy k and flight leg j copy n for an aircraft t.
TA_{base}^{t}	Base turnaround time for an aircraft t .
e_a	The congestion coefficient for the airport a.
S_{ijt}^{kn}	Idle time between flight leg i copy k and flight leg j copy n which are operated successively by the aircraft t.
FT_{max}	Maximum allowable flying time since the last maintenance check.
TO_{max}	Maximum allowable number of take-offs since the last maintenance checks.
MS_{ma}	$MS_{ma} = 1$ if the location of maintenance station m coincides with the airport a and otherwise $MS_{ma} = 0$.
MT	Time needed for a maintenance check.
Q_m	Available man-hours for the maintenance station m.
ET_m	Time when the maintenance station m opens.
OT_m	Time when the maintenance station m closes.
$Daily$	Aircraft daily usage cost.
$Idel$	Unit idle time cost.
$fuel$	Unit fuel cost.
CO_2	Unit CO_2 emission cost.
d	Planning horizon
N	A considerably large number.

Decision variables

σ_{ik} — $\sigma_{ik} = 1$ if the copy k is chosen for the flight leg i and otherwise $\sigma_{ik} = 0$.

$x_{ijtc}^{kn} \in \{0,1\}$ — $x_{ijtc}^{kn} = 1$ if the aircraft t operates flight leg i copy k and flight leg j copy n successively before receiving the cth maintenance check, and otherwise $x_{ijtc}^{kn} = 0$.

$y_{ijtc}^{kn} \in \{0,1\}$ — $y_{ijtc}^{kn} = 1$ if the aircraft t operates flight leg i copy k and flight leg j copy n successively and receives the cth maintenance check in between, and otherwise $y_{ijtc}^{kn} = 0$.

4.2.3 Model formulation

We develop the new mathematical model base on the modified connection network shown in Figure 1. The model is described as below:

$$min \sum_{j\in L}\sum_{n\in N_j}\sum_{t\in T}\sum_{c\in C} Daily\left(x_{ojtc}^{n}\right)d + \sum_{i\in L}\sum_{k\in K_i} \sigma_{ik} FC_{ik}^{AC} + \sum_{i\in L\cup\{o\}}\sum_{k\in K_i}\sum_{j\in L\cup\{s\}}\sum_{n\in N_j}\sum_{t\in T}\sum_{c\in C} S_{ijt}^{kn}(x_{ijtc}^{kn} + y_{ijtc}^{kn})Idle \quad (4.1)$$

$$\sigma_{ik} = \sum_{j\in L\cup\{s\}}\sum_{n\in N_j}\sum_{t\in T}\sum_{c\in C} x_{ijtc}^{kn} + \sum_{j\in L\cup\{s\}}\sum_{n\in N_j}\sum_{t\in T}\sum_{c\in C} y_{ijtc}^{kn} \quad \forall i \in L, \forall k \in K_i \quad (4.2)$$

$$\sum_{k\in K_i} \sigma_{ik} = 1 \quad \forall i \in L \quad (4.3)$$

$$\sum_{j\in L}\sum_{n\in N_j} x_{ojtc}^{n} + \sum_{j\in L}\sum_{n\in N_j} y_{ojtc}^{n} = 1 \quad c=1, \forall t \in T \quad (4.4)$$

$$\sum_{i\in L}\sum_{k\in K_i}\sum_{c\in C} x_{istc}^{k} + \sum_{i\in L}\sum_{k\in K_i}\sum_{c\in C} y_{istc}^{k} = 1 \quad \forall t \in T \quad (4.5)$$

$$\sum_{j\in L\cup\{o\}}\sum_{n\in N_j} x_{jitc}^{nk} + \sum_{j\in L}\sum_{n\in N_j} y_{jitc-1}^{nk} = \sum_{j\in L}\sum_{n\in N_j} x_{ijtc}^{kn} + \sum_{j\in L}\sum_{n\in N_j} y_{ijtc}^{kn} \quad \forall i \in L, \forall k \in K_i, \forall t \in T, \forall c \in C \quad (4.6)$$

$$x_{ijtc}^{kn} \le \sum_{a\in A} AS_{ika} DS_{jna} \quad \forall i,j \in L, \forall n \in N_j, \forall k \in K_i, \forall t \in T, \forall c \in C \quad (4.7)$$

$$y_{ijtc}^{kn} \le \sum_{a\in A} AS_{ika} DS_{jna} \quad \forall i,j \in L, \forall n \in N_j, \forall k \in K_i, \forall t \in T, \forall c \in C \quad (4.8)$$

$$y_{ijtc}^{kn} \le \sum_{a\in A}\sum_{m\in M} AS_{ika} MS_{ma} \quad \forall i,j \in L, \forall n \in N_j, \forall k \in K_i, \forall t \in T, c \in C \quad (4.9)$$

$$AT_{ik} + TA_{ijt}^{kn} - DT_{jn} \le N(1 - x_{ijtc}^{kn}) \quad \forall i,j \in L, \forall k \in K_i, \forall n \in N_j, \forall t \in \quad (4.10)$$

$T, \forall c \in C$

$$AT_{ik} + MT - DT_{jn} \leq N(1 - y_{ijtc}^{kn}) \quad \forall i,j \in L, \forall k \in K_i, \forall n \in N_j, \forall t \in T, \forall c \in C \tag{4.11}$$

$$\sum_{j\in L}\sum_{n\in N_j}\sum_{i\in L}\sum_{k\in K_i} FT_{jn} y_{ijtc}^{kn} + \sum_{j\in L}\sum_{n\in N_j}\sum_{i\in L}\sum_{k\in K_i} FT_{jn} x_{ijtc+1}^{kn} \leq FT_{max} \quad \forall t \in T, \forall c \in C \tag{4.12}$$

$$\sum_{j\in L}\sum_{n\in N_j}\sum_{i\in L}\sum_{k\in K_i} y_{ijtc}^{kn} + \sum_{j\in L}\sum_{n\in N_j}\sum_{i\in L}\sum_{k\in K_i} x_{ijtc+1}^{kn} \leq TO_{max} \quad \forall t \in T, \forall c \in C \tag{4.13}$$

$$\sum_{i\in L}\sum_{k\in K_i}\sum_{j\in L}\sum_{n\in N_j}\sum_{c\in C} y_{ijtc}^{kn} \geq 1 \quad \forall t \in T \tag{4.14}$$

$$\sum_{a\in A}\sum_{m\in M} MS_{ma}\, AS_{ika}\, OT_m - AT_{ik} \leq N(1 - y_{ijtc}^{kn}) \quad \forall i,j \in L, \forall n \in N_j, \forall k \in K_i, \forall t \in T, \forall c \in C \tag{4.15}$$

$$AT_{ik} + MT - \sum_{a\in A}\sum_{m\in M} MS_{ma}\, AS_{ika} ET_m \leq N(1 - y_{ijtc}^{kn}) \quad \forall i,j \in L, \forall n \in N_j, \forall k \in K_i, \forall t \in T, \forall c \in C \tag{4.16}$$

$$\sum_{i\in L}\sum_{k\in K_i}\sum_{j\in L}\sum_{n\in N_j}\sum_{c\in C}\sum_{a\in A} MS_{ma}\, AS_{ika} y_{ijtc}^{kn} \leq Q_m \quad \forall m \in M \tag{4.17}$$

$$\sigma_{ik} \in \{0,1\} \quad \forall i \in L, \forall k \in K_i \tag{4.18}$$

$$x_{ijtc}^{kn}, y_{ijtc}^{kn} \in \{0,1\} \quad \forall i,j \in L, \forall k \in K_i, \forall n \in N_j, \forall t \in T, \forall c \in C \tag{4.19}$$

The objective function, explained by equation (4.1) minimizes the summation of the fuel-burn related costs, the aircraft usage costs, and the idle time costs thus the trade-off between aircraft utilization and fuel-burn related costs can be achieved. In equation (4.1), the fuel-burn related costs FC_{ik}^{AC} are calculated by equation (3.2)

Constraints (4.2) -(4.5) explain the coverage constraints. The requirement that only one copy can be chosen for one flight leg and one flight leg can be operated by exactly one aircraft is defined in constraints (4.2) -(4.3). Constraints (4.4) -(4.5) ensure the start and the completion of an aircraft route. The balance constraint (4.6) is cast to keep the movement of an aircraft throughout the connection network. It depicts two different scenarios. If an aircraft visits a flight leg copy using a non-maintenance arc, the next flight leg copy must be visited using either a non-maintenance arc or a maintenance arc. In another scenario, if an aircraft visits a flight leg copy using a maintenance arc, the next flight leg copy must be visited using a non-maintenance arc.

Constraints (4.7) -(4.11) describe the location and time issues of flight connections. For two copies of two different flight legs, which can be executed successively by the identical aircraft, location constraint (4.7) indicates that the arrival station of the former one must be the same as the departure station of the latter one, while the time constraint (4.10) denotes that the departure time of the latter one must be larger than the summation of the arrival time of former one and the turnaround time. Herein, the turnaround time TA_{ijt}^{kn} is calculated by multiplying the airport congestion coefficient e_a with the aircraft base turnaround time TA_{base}^{t}. If there is a maintenance check in between, the location constraints (4.8)-(4.9) signify that the three places, including the location of the maintenance station, the arrival station of the former one and the departure station of the latter one, must be the same, while the time constraint (4.11) indicates that the departure time of the latter one must be larger than the summation of the arrival time of the former one and the duration of a maintenance check.

Constraints (4.12) -(4.14) define the three maintenance requirements. They denote that an aircraft must receive a maintenance check before (i) accumulating the maximum flying time, defined by constraint (4.12), (ii) accumulating the maximum number of takeoffs, defined by constraint (4.13), and (iii) accumulating the maximum number of days, defined by constraint (4.14). Moreover, a maintenance check must be performed during the maintenance station’s working time, which is ensured by the constraints (4.15) -(4.16). Constraint (4.15) ensures the arrival time at the maintenance station must be larger than the station’s opening time, while constraint (4.16) guarantees that the completion time of a maintenance check must be smaller than the station’s closing time. Lastly, the constraint (4.17) ensures that there is sufficient workforce when an aircraft visits a maintenance station.

At last, we define the decision variables’ domain in constraints (4.18) -(4.19).

4.3 An IACO algorithm

The following notations required for the ACO algorithm will be defined:

$Attr_{ikjn}$	The attraction of arc (ik, jn) for an ant
q_0	A predefined parameter $(0 < q_0 < 1)$
q	A random number (which is uniformly distributed in the range of 0 to 1)
ρ	The parameter that controls the evaporation rate $(0 < \rho < 1)$
τ_{ikjn}	The pheromone trail deposited on arc (ik, jn)
η_{ikjn}	The heuristic information on arc (ik, jn)
φ_{jn}	The heuristic information on node jn
α, β, γ	The parameter to adjust the relative influence of τ_{ikjn}, η_{ikjn}, and φ_{jn}
$\Delta\tau_{ikjn}^{best}$	The pheromone increments on arc (ik, jn) which is contained in the current best-found solution at each iteration
N_{ik}^{lv}	Set of feasible nodes which can be connected with node ik

The ACO, as originally proposed by Dorigo, was encouraged by the behavior of real ant colonies in search of food. An ant deposits one special type of secretion, called pheromone, on the way from their nest to food source, and in turn this pheromone trail is used to guide the successor ants. By this indirect but useful communication form, positive feedback can be achieved, and thereby the shortest path from the nest to the food resource can be found. In the ACO algorithm, artificial ants mimic this foraging behavior. They walk on a weighted and constrained graph, over which the combinatorial optimization problem is modeled. Then they identify the optimal paths based on the pheromone trails that they laid to record the promising searching space, and the problem-dependent heuristic information.

In our study, an IACO algorithm is developed. Next, we will explain the main elements of the proposed algorithm.

4.3.1 State transition mechanism

When an ant is on the node ik, the following state transition mechanism will be utilized to select the next node jn.

$$f(x) = \begin{cases} arg\ \_max_{jn \in N_{ik}^{lv}} \{Attr_{ikjn}\}, if\ q \leq q_0 \\ p_{ikjn}^{lv}, \qquad\qquad\qquad if\ q > q_0 \end{cases} \tag{4.20}$$

When q is less than the predefined parameter q_0, the ant will choose the node jn with the biggest $Attr_{ikjn}$ as the next node. Otherwise, the ant will randomly select the next node according to the probability p_{ikjn}^{lm} defined by the formula below:

$$p_{ikjn}^{lv} = Attr_{ikjn} \Big/ \sum_{dw \in N_{ik}^{lv}} Attr_{ikdw} \qquad if\ jn \in N_{ik}^{lv} \tag{4.21}$$

And the attraction value of arc (ik, jn) for an ant is

$$Attr_{ikjn} = [\tau_{ikjn}]^{\alpha} [\eta_{ikjn}]^{\beta} [\varphi_{jn}]^{\gamma} \tag{4.22}$$

In the conventional ACO algorithm, an ant incrementally construct a solution on the graph, where all the nodes must be visited at least once and only once. Accordingly, only the attractiveness of arcs is considered in the state transition rule and the sequencing is the only issue to be concerned. However, in our study, due to the existence of the flexible cruise time, there are several copies with different cruise times for each flight leg and only one copy is allowed to be chosen, implying that not all the nodes on the modified connection network are to be covered. The node selection is also crucial. Therefore, the heuristic information of the node jn (φ_{jn}) is introduced as a guide for node selection. Then, we will address the τ_{ikjn}, η_{ikjn} and φ_{jn} in detail:

- ***Pheromone information of arc (ik, jn): τ_{ikjn}*:** The pheromone information represents the ants' search experience and directs the future ants to search in the promising space. Traditionally, only one pheromone structure is involved as all arcs are of the same character. However, the modified connection network contains two distinct types of arcs including non-maintenance arcs and maintenance arcs, so as to comply with mandatory maintenance requirements. Consequently, two pheromone structures are set for two types of arcs, thus allowing for more accurate representation of search experience.

- ***Heuristic information of arc*** $(\boldsymbol{ik,jn})$***:*** $\boldsymbol{\eta_{ikjn}}$**:** The heuristic information is usually based on the characteristics of the problem. Since the objective of this study is to minimize the number of required aircraft, the arcs with less idle time are preferred and consequently have a higher heuristic value. Therefore, the heuristic value is inversely proportional to the value of idle time S_{ijt}^{kn}. Moreover, when an ant is positioned on the node with cruise time compression, it is reasonable to give the priority to the arcs that are only feasible after this cruise time compression. In the new heuristic function, these arcs are distinguished from the originally feasible arcs and their heuristic values are doubled. The addition of one is utilized to avoid division by 0. The heuristic function is defined as below:

$$\eta_{ikjn} = \begin{cases} 1/(\sqrt{S_{ijt}^{kn}} + 1)\ if\ arc(ik,jn)\ is\ the\ originally\ connected\ arcs \\ 2/(\sqrt{S_{ijt}^{kn}} + 1) \qquad otherwise \end{cases} \tag{4.23}$$

- ***Heuristic information of node jn:*** $\boldsymbol{\varphi_{jn}}$ **:** In the list of feasible nodes to be next covered, the nodes of the same flight leg share the same heuristic information of arc (ik,jn). However, they have different cruise times, resulting in different flight connection opportunities that have direct impact on the route construction. The node with improved flight connection opportunities can bring more efficient flight connections and thus be more preferred. Therefore, the heuristic information of node jn (φ_{jn}, equation (4.24)) is introduced.

$$\varphi_{jn} = FCO_{jn}/FCO_{jr} \tag{4.24}$$

Where the FCO_{jn} and the FCO_{jr} denote the number of nodes that node jn and node jr can connect, respectively. Node jr represents the copy r of flight leg j that doesn't have cruise time compression.

4.3.2 The global pheromone trail update

After each iteration, the pheromone trail on the arcs is updated using the following equation:

$$\tau_{ikjn} \leftarrow (1-\rho)\tau_{ikjn} + \Delta\tau_{ikjn}^{best} \tag{4.25}$$

where the first term reduces the amount of pheromone on all the arcs, and the second term gives the pheromone increments on the arcs of the current best-found solution. The pheromone increments are calculated using the formula below:

$$\Delta\tau_{ikjn}^{best} = A \times B \times C \times (Q/cost(R_{best})) \tag{4.26}$$

$$A = e^{-5((m+n)/(M+N))} \tag{4.27}$$

$$B = 1/\left(1 + 0.95^{i}\right) \tag{4.28}$$

$$C = N \tag{4.29}$$

In the conventional ACO, the pheromone increments on the arcs included in the current best-found solution are set to be $Q/cost(R_{best})$. $cost(R_{best})$ is the best objective value found from the start till the present iteration. Q is the control parameter that is used to adjust the amount of deposited pheromone and thus helps the algorithm avoid local optimal convergency or excessively random search. To increase search efficiency and precision, three multipliers A, B and C are applied to the term of pheromone increments. In the following, we will introduce these three multipliers in detail.

- ***Multiplier A*:** As was mentioned and proved in the study of Naimi and Taherinejad (2009), giving all the arcs the same pheromone increments is not a good strategy for pheromone updating. At the initial stage of route construction, an ant has greater freedom in choosing the next flight leg to be covered since few flight legs have been covered and are prohibited to be chosen. In contrast, at the final stage, most of the flight legs have been covered which strongly constrains the available flight legs. Accordingly, a poor decision made at the initial stage may lead to a series of poor choices and, eventually, a disastrous solution. Therefore, it is rational to allow up-to-now best ant to have greater impact on pheromone updating at its initial stage of a tour and less impact when it is about to finish its tour. Based on above discussion, the multiplier A is proposed. In this equation (4.27), M is the number of routes in the solution, m denotes the mth route that the arc $(ik,\ jn)$ is in, N is the number of nodes covered in mth route, and n signifies the number of nodes covered before choosing arc $(ik,\ jn)$ in mth route. Apparently, both m and n increase from the beginning to the ending of a solution. Therefore, the multiplier A decreases towards zero ($e^{-5} \approx 0$) , accurately reflecting the gradually diminishing role of the up-to-now best ant in the pheromone updating from the beginning to the ending of a tour.

- ***Multiplier B:*** During the process of iterations, the solutions' quality will be greatly improved, and the ants will gradually shift their focus from the exploration to the exploitation. Therefore, fewer pheromone increments should be applied to the earlier iterations to increase the solutions' diversity and avoid a rapid fall into the local optimal, while more pheromone increments should be applied to the later iterations to strengthen the exploitation and accelerate the convergency. To achieve this purpose, multiplier B (equation (4.28)) is introduced, where i denotes the iteration index. This multiplier increases the pheromone increments during the process of iterations, allowing for dynamically adjusting the trade-off between exploration and exploitation so as to improve the search efficiency.

- ***Multiplier C:*** The routes covering more flight legs are more favorable due to its higher aircraft utilization. Thus, it is logical to allocate more pheromone increments to the route covering more flight legs, which enables the subsequent ants to perform more delicate searches in a more favorable area in the next iteration. Therefore, we propose the multiplier C, which is denoted by N the number of nodes covered by the mth route where the arc $(ik,\ jn)$ is.

4.3.3 The proposed algorithm

The overview of the proposed algorithm is illustrated in Figure 4.2 and the detailed algorithm steps are specified below:

Step 1: Initialize the ACO parameters including $\alpha, \beta, \gamma, \rho, q_0, Q$, and the maximum number of iterations.

Step 2: Create the aircraft list T.

Step 3: Store the copies of all flight legs in the list FN.

Step 4: Constructing the route by completing the sub-steps listed below.

Step a: Examine whether the list FN is empty. If it is, perform *step 6*, otherwise perform *step b*.

Step b: Select an aircraft t from the list T to represent an ant and start route construction on the source node.

Step c: Scan the list FN and choose the first copy k of flight leg i whose departure time is the earliest.

Step d: For the selected copy k of flight leg i, delete all copies of flight leg i in the list FN and insert the copy k of flight leg i into the constructed route for the aircraft t.

Step e: Choose the next possible flight copies considering the location and time constraints (4.7) and (4.10). If no possible copies of flight legs exist, perform *step l,* otherwise perform *step f.*

Step f: For the possible copies of flight legs, examine whether they satisfy the maintenance constraints (4.12) -(4.14). Store the maintenance feasible copies of the flight legs in the list PF. If the list PF is empty, perform *step g*, otherwise perform *step i*.

Step g: Receive a maintenance check at the maintenance station considering the location constraint (4.9), the working times and the capacity constraints (4.15), (4.16) and (4.17).

Step h: Once finishing the maintenance check, choose the next possible copies of flight legs considering location and time constraints (4.8) and (4.13). Store the possible copies of the flight legs in the list PF. If the list PF is empty, perform *step l*, otherwise perform *step* i.

Step i: Use the state transition mechanism defined in equation (4.20) and (4.22) to select the copy n of flight leg j from the list PF.

Step j: For the selected copy n of flight leg j, delete all copies of flight leg j in the list FN and insert the copy n of flight leg j into the constructed route for the aircraft t.

Step k: Examine whether the list FN is empty. If it is, end the route on the sink node and perform *step 6*, otherwise perform *step* e.

Step l: End the route on the sink node and move to *step* a.

Step 5: Compare the solution with the current best-found solution and make an update if necessary.

Step 6: Use equation (4.25) to update the pheromone trails.

Step 7: Examine the stopping criteria. If it is satisfied, terminate the algorithm, otherwise increase the iteration number, and move to step 2.

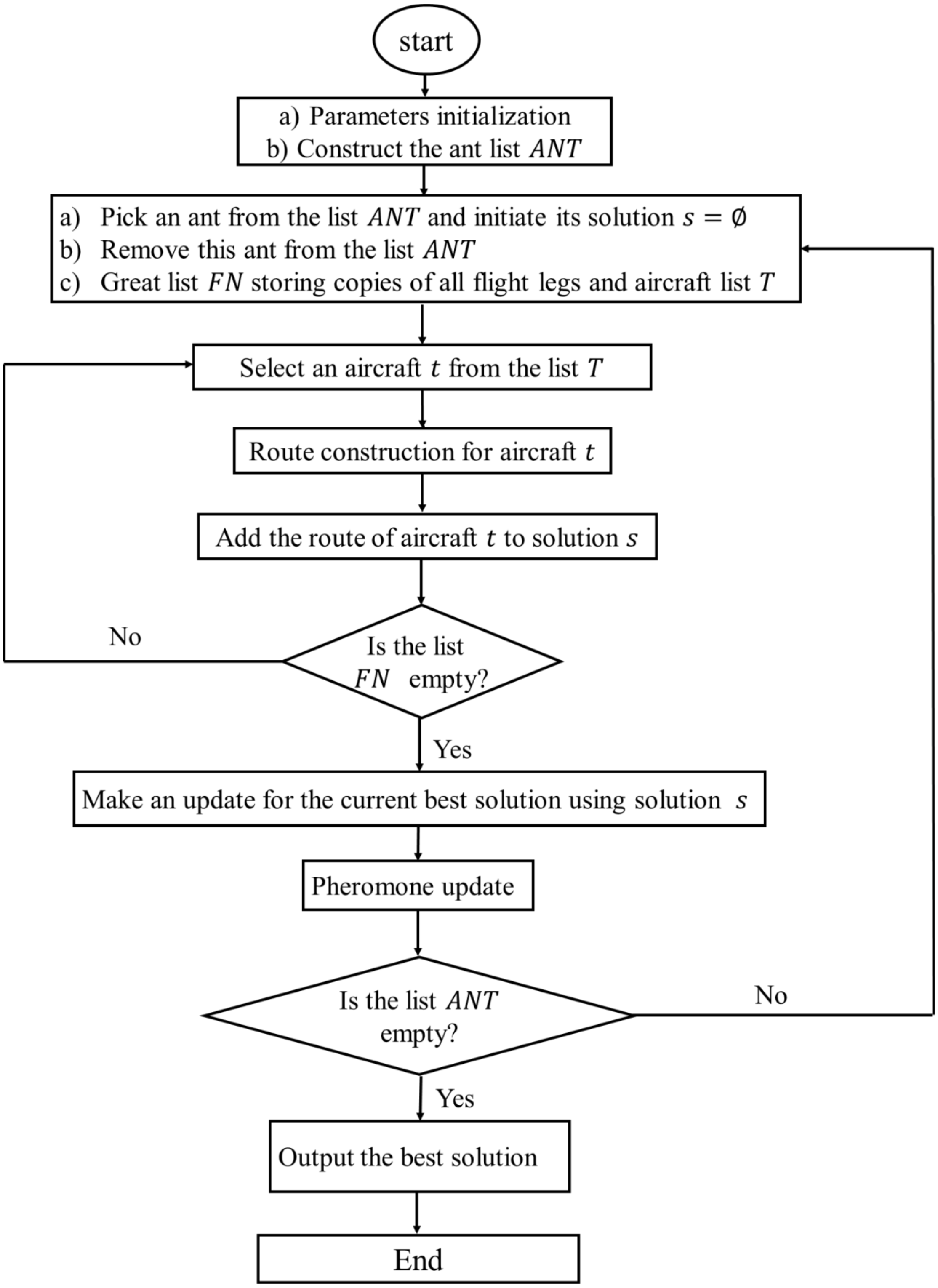

start
a) Parameters initialization
b) Construct the ant list ANT
a) Pick an ant from the list ANT and initiate its solution $s = \emptyset$
b) Remove this ant from the list ANT
c) Great list FN storing copies of all flight legs and aircraft list T
Select an aircraft t from the list T
Route construction for aircraft t
Add the route of aircraft t to solution s
Is the list FN empty?
No
Yes
Make an update for the current best solution using solution s
Pheromone update
Is the list ANT empty?
No
Yes
Output the best solution
End

Figure 4.2: The overview of the proposed algorithm

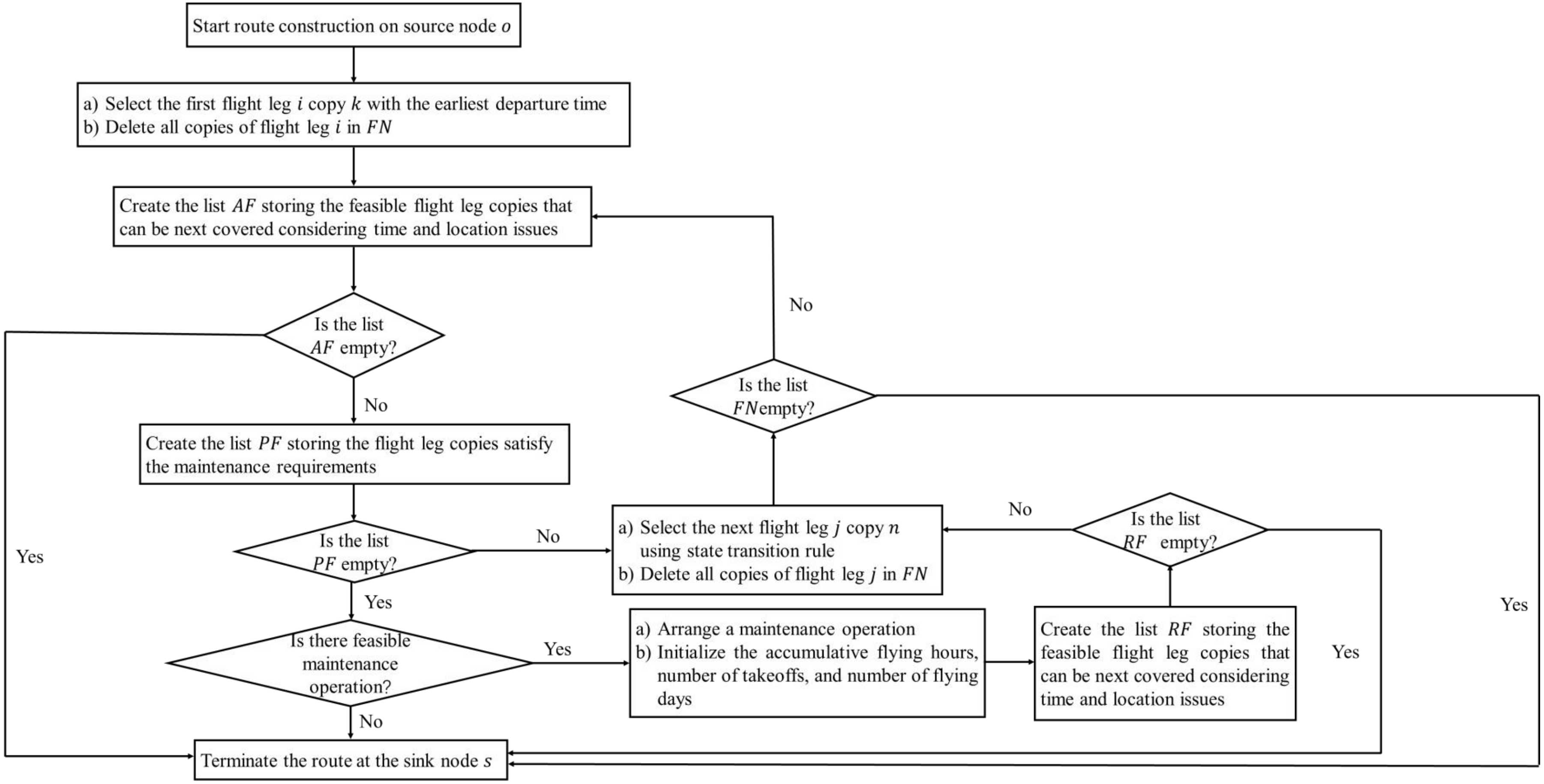

Start route construction on source node o
a) Select the first flight leg i copy k with the earliest departure time
b) Delete all copies of flight leg i in FN
Create the list AF storing the feasible flight leg copies that can be next covered considering time and location issues
Is the list AF empty?
No
Yes
Create the list PF storing the flight leg copies satisfy the maintenance requirements
Is the list PF empty?
No
Yes
Is there feasible maintenance operation?
Yes
No
Terminate the route at the sink node s
a) Select the next flight leg j copy n using state transition rule
b) Delete all copies of flight leg j in FN
Is the list FNempty?
No
Yes
a) Arrange a maintenance operation
b) Initialize the accumulative flying hours, number of takeoffs, and number of flying days
Create the list RF storing the feasible flight leg copies that can be next covered considering time and location issues
Is the list RF empty?
No
Yes

Figure 4.3: The steps of route construction in the proposed algorithm

4.4 Computational experiments

4.4.1 Test instance and experimental setup

In the experiments, we collect ten cases of flight schedules from the Bureau of Transportation Statistics (BTS, 2020) database. Table 4.1 presents the main characteristics of the collected test cases. The ten test cases are involved with an increasing size with 40, 83, 120, 150, 222, 260, 312, 410, 500 and 613 flights, respectively.

For all test cases, based on airlines' recommendation, we assume that maximum allowed flying hours are 40 hours, the maximum allowed take-offs are 8, and the duration of the maintenance check is 6 hours. According to the study of Duran et al. (2015), we assume that the unit CO_2 emission cost is \$0.02/kg, the scheduled non-cruise time is 20 minutes, the cruise time window is $[85\% \times original\ cruise\ time, original\ cruise\ time]$, and the compression interval is $5\% \times original\ cruise\ time$. Referring to the IATA fuel price monitor, the unit fuel cost is assumed to be \$1.26/kg. The most representative aircraft type A320 111, of which the detailed information is provided in the Table 3.2, is considered in this study. Its daily usage costs are taken as $Idle \times 600$.

Before adopting the IACO algorithm, a parameter tunning process is necessary due to the direct impact of the parameter on the performance of IACO algorithm. By using the Taguchi method, the best parameters are set as $\alpha = 1, \beta = 1, \gamma = 1\ q_0 = 0.95, \rho = 0.15$, and number of ants =fleet size. To examine the average performance, the algorithm runs 30 times for each test case since no better result could be obtained by increasing the number of runs.

Table 4.1: Test cases

Test cases	Flight legs	Fleet size	Airports	Maintenance stations
1	40	8	14	7
2	83	11	24	10
3	120	15	38	16
4	150	15	42	16
5	222	21	53	18
6	260	25	52	18
7	312	34	56	18
8	410	45	57	21
9	500	43	54	21
10	613	62	57	21

4.4.2 Performance comparison of IACO with the CPLEX solver

In this section, the CPLEX solver solves the MILP model directly. However, it is challenging for the CPLEX solver to solve the medium and large-size cases optimally due to the significant increase in the computation time. Therefore, the exact solution derived by the CPLEX solver is utilized as a criterion for the small-size test cases, while the upper bound obtained from the CPLEX solver is used as a comparison for the medium and large-size test cases. Note that the upper bound is achieved within 7 hours run time limitation of the CPLEX solver.

4.4.2.1 Small-size problems analysis

Generally, the schedules with the number of flight legs below 250 are recognized as small-size cases for most airlines (Ruan et al. 2021; Eltoukhy et al. 2018a). Referring to Table 4.2 (which shows the number of remaining copies of flight legs after the pre-processing step), we consider the first five test cases in Table 4.1 as small-size cases. Table 4.3 reports the comparison between the solutions provided by the CPLEX solver, and that obtained from the IACO algorithm. The first column indicates the test case number. The next two columns show the solutions obtained from the CPLEX solver, where R^* column and T column represent the exact solution and the computation time, respectively. In the remaining columns of Table 4.3, the solutions derived by the proposed algorithm are presented. R_{best} column, R_{ave} column, σ column and $\bar{T}$ column describe the best solution, the average solution, the standard deviation, and the average computation time, respectively. The %Gap column records the percentage difference between the optimal solution and the average solution, which is calculated using the formula (4.30).

$$Gap = (R_{ave} - R^*)/R^* \tag{4.30}$$

From Table 4.3, it can be observed that for the first test case the values of both R_{ave} and R_{best} are the same with the value of R^*. For the other four test cases, when the size of test cases increases, the values of R_{best} still equals to the values of R^* while the values of R_{ave} slightly differ from the values of R^* by no more than 0.42%. From σ column, it shows that there is no variance in the solutions obtained from the IACO algorithm in the first test case. The variance slightly increases with the increasing size of the test cases, as shown in the last four test cases. These results well demonstrate the accuracy and the robustness of the ICAO algorithm.

Regarding the computation time, the T and $\bar{T}$ column show that the proposed algorithm can achieve the solutions within 70 seconds whereas the CPLEX solver requires up to 3 hours. Apparently, it takes much less time for the IACO algorithm in comparison with the CPLEX solver. Moreover, the computation time greatly increases with the increasing size of test cases for the CPLEX solver while there is no significant variation in the computation time for the IACO algorithm.

To summarize, concerning the computation time, the IACO algorithm can achieve the significant computation time reduction in comparison with the CPLEX solver. Concerning the solution quality, the IACO algorithm can yield good-quality solutions as the best solutions are the same with the optimal solutions and the average solutions differ from the optimal solutions by no more than 0.42%.

4.4.2.2 Medium and large-size problems analysis

For the purpose of evaluating the capability of the IACO algorithm to tackle the real-world problems, the IACO algorithm is tested on the medium and large-size problems (which denote the last five test cases in Table 4.1). Table 4.4 summaries the comparison results, which contains similar information as Table 4.3 except for the UB column. The UB column denotes the upper bound provided by the CPLEX solver within the time limitation.

From the Table 4.4, we can observe that the IACO algorithm still significantly outperforms the CPLEX solver. Regarding the solution quality, the values of R_{best} are identical to the values of UB in all test cases, whereas the values of R_{ave} differ from the values of UB by no more than 12%. Moving to computational time, albeit the considerably enlarged network, our algorithm can still solve it within 15 minutes, which is reasonable and practical. Moreover, taking a close look at the σ, although the values increase, they imply the small solution variances, as the values of solutions significantly increase. This confirms the robustness of the IACO algorithm in dealing with the medium and large-size problems.

Table 4.2: Number of nodes comparison

Test cases	Flight legs	Remaining nodes
1	40	46
2	83	96
3	120	133
4	150	164
5	222	238
6	260	282
7	312	338
8	410	461
9	500	579

10	613	683

Table 4.3: Performance comparison with CPLEX for small-size cases

Test cases	**CPLEX**		**IACO**				**%Gap**
	R^*	$T(s)$	R_{best}	R_{ave}	σ	$\bar{T}(s)$	
1	942,453	10.43	942,453	942,453	0	4.58	0
2	5,696,567	78.62	5,696,567	5,702,683	2,031	10.57	0.11
3	12,098,788	302.31	12,098,788	12,128,159	8,728	20.79	0.24
4	5,850,835	472.45	5,850,835	5,863,069	2,936	28.07	0.21
5	8,705,534	10215.46	8,705,534	8,741,893	18,749	64.22	0.42

Table 4.4: Performance comparison with CPLEX for medium and large-size cases

Test Cases	**CPLEX**	**IACO**				**%Gap**
	UB	R_{best}	R_{ave}	σ	$\bar{T}(s)$	
6	18,461,505	18,461,505	19,175,821	171,221	98.52	3.73
7	13,268,536	13,268,536	13,804,348	392,180	167.94	4.04
8	16,030,559	16,030,559	17,573,417	558,061	328.55	9.62
9	33,141,047	33,141,047	34,095,754	740,870	526.82	2.88
10	23,782,212	23,782,212	26,415,509	1,073,948	796.09	11.07

4.4.3 Performance comparison of IACO with three meta-heuristic approaches

In this section, we solve the proposed model using three meta-heuristic approaches, i.e., convention ACO, SA, and GA, under the same experimental conditions. The parameters settings are identical for the conventional ACO and the ICAO. For GA and SA, we applied the identical procedures and parameters introduced by Eltoukhy et al (2017b). The performance of the three meta-heuristic approaches is summarized in Table 4.5, and the percentage improvement of IACO over these three approaches in solution quality is illustrated in Table 4.6 and Figure 4.4 to 4.5. $IMP\%$ is computed by $(R_{CAL} - R_{IACO})/R_{IACO}$ where R_{IACO} and R_{CAL} represent the solution of the IACO and the solution of the algorithm that is compared with the IACO, respectively. Additionally, the computation time comparison is displayed in Table 4.7 and Figure 4.6 to 4.8.

From Table 4.5, it can be interpreted that conventional ACO algorithm can produce considerably superior solutions in comparison with SA and GA, and even be capable to generate the exact solution for the particularly small-size case such as test case 1. Moreover, its computation time slightly increases in some of test cases as compared to SA but is far less than that of GA for all test cases, which can be clearly seen in Table 9. For instance, in test case 10, the computation time of GA is up to 1 hour whereas that of the other two is around 10 minutes. Although SA consumes less time to produce the solutions, the solution quality is the poorest. In case 1, the solution obtained by SA significantly deviates from the optimality, even up to 130.25%.

Moving to the IACO algorithm, we can see from Table 4.3 to 4.4 that the average gap of IACO with the exact solution for small-size test cases is within 0.5%, while that with the upper bound is within 12% for medium and large- size test cases, which is somewhat less in comparison with the other three approaches, particularly SA and GA. Moreover, its remarkable outperformance in terms of the solution quality is visible in Table 4.6 and Figure 4.4 to 4.5. Along with the significant improvement in solution quality, the computation time of the IACO algorithm is slightly longer than the conventional ACO and SA for the medium and large-size test cases, while in some of small-size test cases, such as case 2,3 and 4, the computation time is even shorter. This slight increase in computation time is worthwhile in comparison with the considerable improvement in solution quality. Taking case 7 as an example, 32 seconds increase in computation time can improve the average solution quality by around 39.79% in comparison with SA, while 21 seconds increase in computation time can result in 16.98% improvement in average solution quality as compared to the conventional ACO.

In the following, we will explain the rationalities behind the above results.

In the AMRP incorporating with cruise speed control, different from the traditional AMRP, there are several copies with different cruise times for each flight leg, and only one copy can be chosen. In SA, it starts with an initial solution in which the copy for each flight leg has been determined, and then improves the current solution by local search in which the copy for each flight leg has been fixed. As a result, the cruise speed control only plays the role in the initial route construction. This significantly diminishes the positive impact of cruise speed control on route construction, resulting in great reduction in the solution space. Thus, the solutions obtained by SA are the poorest. In the GA, the solution quality is improved based on a population of solutions. The solution space is enlarged as compared to SA due to more choices of copies for each flight leg, and better solutions can be produced.

However, in the conventional ACO algorithm and the IACO algorithm, they construct a new solution from an empty solution. The copy for each flight leg dynamically changes at each time of route construction, which fully utilizes the flexible cruise time. Therefore, their solutions are far better as compared to SA and GA. While comparing the conventional ACO algorithm and the IACO algorithm, the solution quality of IACO algorithm can be greatly improved without significant time variance due to the efficacy of the new strategies.

Based on the above discussions, we can conclude that the IACO algorithm notably outperforms other three algorithms regarding the solution quality while the computation time slightly increases as compared to SA and the conventional ACO in the medium and large-size test cases. Hence, the efficiency of the IACO algorithm has been verified.

Table 4.5: Performance of the other three approaches

Test cases	UB	**The conventional ACO**			**SA**			**GA**		
		R_{best}	R_{ave}	**%Gap**	R_{best}	R_{ave}	**%Gap**	R_{best}	R_{ave}	**%Gap**
1	942,453	942,453	942,453	0	2,169,998	2,169,998	130.25	1,653,063	1,653,063	75.40
2	5,696,567	5,925,061	5,931,915	4.13	8,358,922	8,423,818	47.88	8,226,557	8,228,362	44.44
3	12,098,788	12,475,909	12,490,795	3.24	14,003,494	14,166,171	17.09	13,960,910	13,980,505	15.55
4	5,850,835	6,084,835	6,166,776	5.40	9,151,499	9,216,232	57.52	8,462,961	8,489,085	45.09
5	8,705,534	9,540,768	9,557,255	9.78	11,979,579	12,023,937	38.12	11,267,337	11,312,610	29.95
6	18,461,505	20,712,426	20,865,974	13.02	22,120,281	22,448,473	21.60	21,234,926	21,677,603	17.42
7	13,268,536	15,843,229	16,148,909	21.71	19,206,652	19,296,554	45.43	18,474,716	18,893,036	42.39
8	16,030,559	17,750,186	18,610,041	16.09	24,368,613	25,163,323	56.97	22,112,424	23,250,217	45.04
9	33,141,047	37,166,883	37,672,274	13.67	39,372,149	40,677,344	22.74	38,101,892	39,213,505	18.32
10	23,782,212	28,044,151	29,178,249	22.69	31,057,124	35,822,348	50.63	30,030,487	33,571,978	41.16

Table 4.6: Solution quality comparison with three metaheuristic approaches

Test cases	Average solution (*IMP%*)			Best solution (*IMP%*)		
	The conventional ACO	**SA**	**GA**	**The conventional ACO**	**SA**	**GA**
1	0.00%	130.25%	75.40%	0.00%	130.25%	75.40%
2	4.02%	47.72%	44.29%	4.01%	46.74%	44.41%
3	2.99%	16.80%	15.27%	3.12%	15.74%	15.39%
4	5.18%	57.19%	44.79%	4.00%	56.41%	44.65%
5	9.33%	37.54%	29.41%	9.59%	37.61%	29.43%
6	8.81%	17.07%	13.05%	12.19%	19.82%	15.02%
7	16.98%	39.79%	36.86%	19.40%	44.75%	39.24%
8	5.90%	43.19%	32.30%	10.73%	52.01%	37.94%
9	10.49%	19.30%	15.01%	12.15%	18.80%	14.97%
10	10.46%	35.61%	27.09%	17.92%	30.59%	26.27%

Table 4.7: Computation time comparison with three metaheuristic approaches

Test cases	Computation time (s)			
	The conventional ACO	**SA**	**GA**	**IACO**
1	4.85	3.83	12.12	4.58
2	11.51	12.63	36.54	10.57
3	27.82	29.23	66.13	20.79
4	34.67	32.97	92.64	28.07
5	66.19	64.18	448.98	64.22
6	92.55	88.04	523.74	98.52
7	146.47	135.32	908.98	167.94
8	301.98	296.42	1424.32	328.55
9	464.45	470.62	2917.51	526.82
10	735.76	601.91	3375.29	796.09

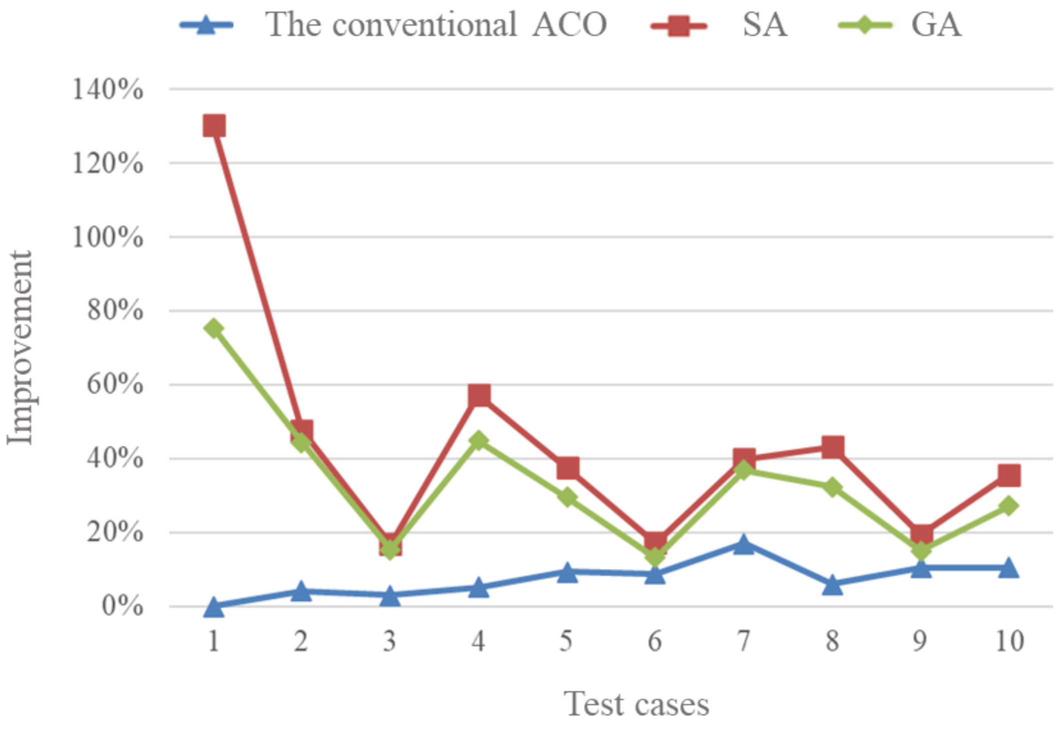


Figure 4.4: Improvement in average solution over the other three approaches

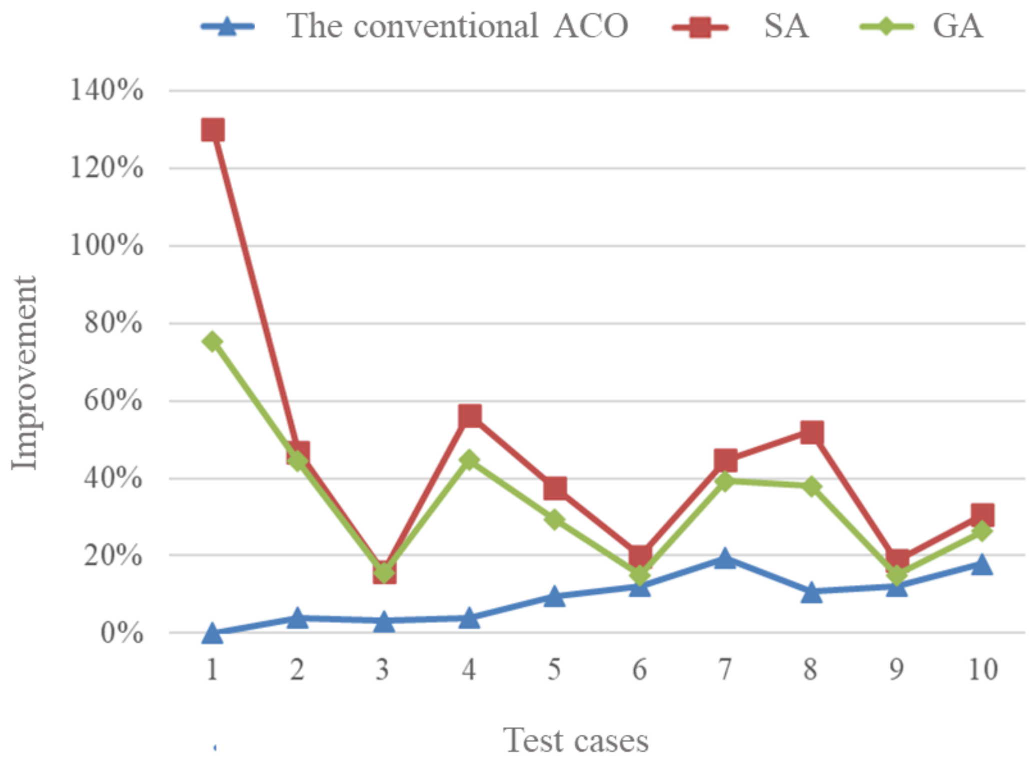


Figure 4.5: Improvement in best solution over the other three approaches

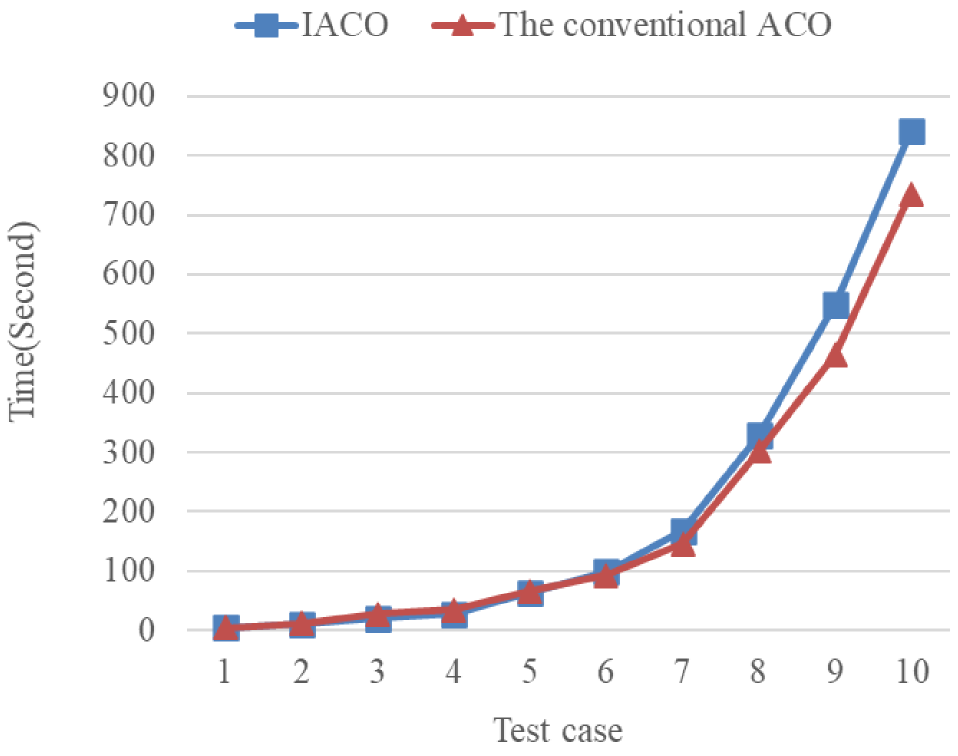


Figure 4.6: The computational time comparison with the conventional ACO

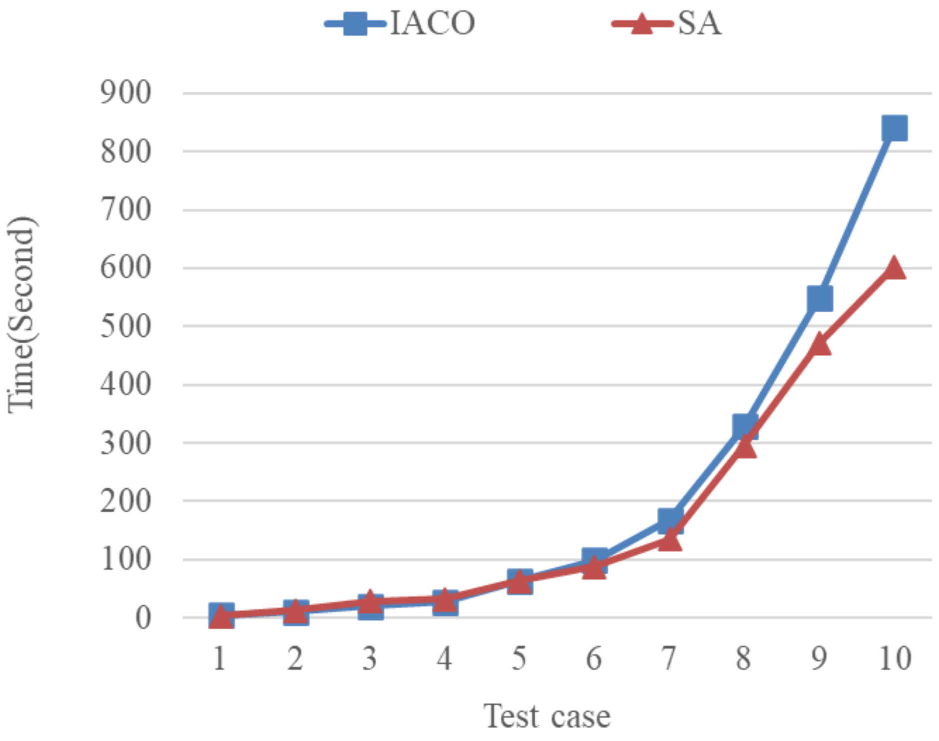


Figure 4.7: The computational time comparison with SA

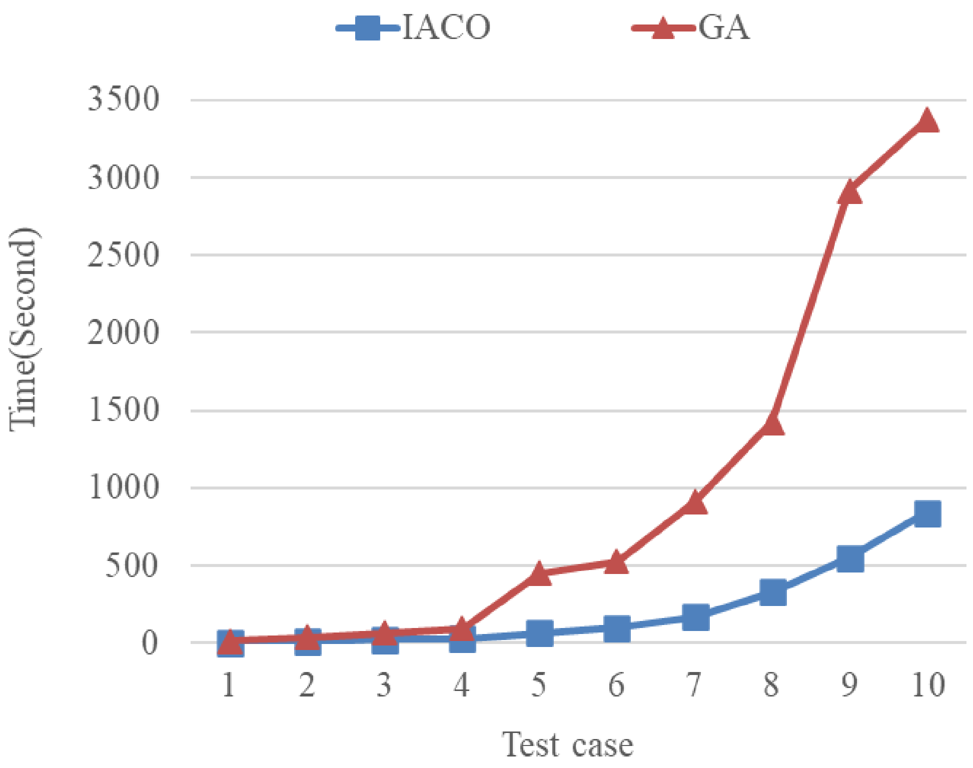


Figure 4.8: The computational time comparison with GA

4.5 Summary

This chapter proposed an IACO algorithm to deal with the AMRP incorporating cruise speed control. The traditional ACO chooses the next covered flight leg depending on the attractiveness of flight connections, thus the sequence of flight legs is the only concerned issue. However, with the addition of the flexible cruise time, the procedure of solution construction involves two steps simultaneously: (i) selecting the cruise times and (ii) determining the aircraft maintenance routes. The selection of cruise times is critical in finding optimal routes thus achieving aircraft savings. To guide the selection of cruise times, a new state transition mechanism incorporates the node-based heuristic information regarding the flight connection opportunities. In addition, the performance of the traditional ACO decreases dramatically in tackling large-scale problems, particularly when modeling flexible cruise time causes an explosion in problem size. To tackle this problem, a modified pheromone updating rule is designed to enhance the search efficiency and precision. Moreover, the pheromone structure and the heuristic function are modified to be more problem specific.

To validate the performance of IACO algorithm, two computational experiments are carried out based on data sets derived from the BTS. First, the solutions derived by the IACO algorithm and the CPLEX solver are compared. From the results analysis, one can say that the proposed algorithm can achieve substantial computation time savings and make a good compromise between the solution quality and the computation time in comparison with the CPLEX solver. Second, the solutions obtained from the proposed algorithm and three promising meta-heuristic approaches, including the conventional ACO, SA, and GA, are compared. The results reveal that the IACO algorithm remarkably surpasses the three meta-heuristic approaches in the solution quality while the computation time is slightly longer in medium and large-size test cases, which is acceptable, compared with the conventional ACO and SA.

Chapter 5 - A Matheuristic Approach for Aircraft Maintenance Routing Problem incorporating Cruise Speed Control (AMRP-CSC)

5.1 Introduction

More recently, a new hybrid optimization method combining exact methods and metaheuristics to exploit complementary strengths of both, called matheuristic, has emerged (Moussavi et al. 2019; Machado et al. 2021). There is no consolidated structure for this method and the hybridization can be very flexible depending on the specific properties of the problem. The increasing maturity of this area promotes the widespread and successful application of matheuristic in the field of transportation, particularly in the vehicle routing problem (VRP) (Wang et al. 2017; Leggieri and Haouari 2018; Queiroga et al. 2021). For tackling variants of VRP, the matheuristic frameworks go to three main directions: set-partitioning/set-covering formulations based, local branching approaches based, and decomposition approaches based (Doerner and Schmid 2010). Thereinto, different set-covering/partitioning-based approaches have successfully solved various VRPs, for instance, the capacitated VRP (Wang et al. 2017; Leggieri and Haouari 2018; Queiroga et al. 2021), the multi-depot VRP (Mancini 2016; Bertazzi et al. 2019), and the VRP with time window (Kramer et al. 2015). The core idea of these approaches is that a meta-heuristic creates a pool of routes, and then a set-covering/partitioning procedure improves the solution by extracting the best combination of routes in the pool. As we know, the AMRP and the VRP are strongly related since both of them can be reduced to several TSPs. However, the researchers have not devoted much attention to the application of matheuristic in the AMRP. In this chapter, a matheuristic approach is developed for AMRP-CSC that combines the metaheuristic part based on the IACO algorithm presented in chapter 4 and the exact components including a set partitioning (SP) procedure and a neighborhood search (NS) procedure. This matheuristic approach is analyzed and tested using the data extracted from the Bureau of Transportation Statistics (BTS), and then its accuracy and the efficiency have been demonstrated by experiment analyses.

The rest of this chapter is arranged as follows. Section 5.2 describes the three main components and the hybridization schemes of the matheuristic approach. The computational experiments are reported in section 5.3. Finally, this chapter is concluded in section 5.4.

5.2 A matheuristic approach for AMRP-CSC

In this section, a matheuristic approach is proposed to solve the problem. Since the meta-heuristic part, i.e., IACO algorithm, is introduced in chapter 4, Section 5.2.1and 5.2.2 outline the exact parts including a SP procedure and a NS procedure. Then, how to merge them to efficiently tackle the AMRP-CSC will be described.

5.2.1 A SP procedure

In IACO, a solution can be rejected due to its worse objective value in comparison with the current best solution. However, a rejected solution could also contain good routes, which can be used to form a better solution by matching the routes of different solutions. In this regard, we populate a pool of routes using the solutions derived from the IACO algorithm. By solving a SP model with the columns corresponding to the routes in the pool, an improved solution was produced. The SP model is described as below.

Let Ω be the pool of routes. We define z_r as binary variable that takes value of 1 if a route $r \in \Omega$ is selected, c_r as its costs and α_{ikr} as a binary parameter that takes value of 1 if fight leg $i \in F$ copy $k \in K_i$ is covered by route $r \in \Omega$. Consider the following SP formulation:

$$min \sum_{r \in \Omega} z_r \, c_r \tag{5.1}$$

$$\sum_{r \in \Omega} \sum_{k \in K_i} z_r \alpha_{ikr} = 1 \quad i \in F \tag{5.2}$$

$$z_r \in \{0,1\} \qquad r \in \Omega \tag{5.3}$$

The objective function (5.1) minimizes the summation of costs by selecting the best combination of routes. Constraint (5.2) states that each flight leg $i \in F$ must and only be covered once.

5.2.2 A SN procedure

This phase is to explore the neighborhoods of the incumbent best-solution by iteratively solving the reduced AMRP-CSC instances. Each reduced AMRP-CSC instance is formulated as a MILP model and is solved by a commercial solver. This section will describe the MILP model first, and then specify the NS procedure in detail.

5.2.2.1 A MILP model

Here, we establish the connection network for the problem, and formulate the MILP model. The notations used though-out this section are defined as follows.

$i, j \in L$	Set of flight legs
$n \in N_i, m \in N_j$	Copy set for flight leg $i, j \in L$
$k \in K$	Set of aircraft
$ms \in MS$	Set of maintenance stations
DT_i	The departure time of flight leg i
AT_{in}	The actual arrival time of flight leg i copy n
CT_{in}	The actual cruise time of flight leg i copy n
FT_{in}	The actual flying time of flight leg i copy n
MT	The time required A-check.
SC_i	The scheduled cruise time of flight leg i
$[SC_i, SC_i(1-\gamma)]$	The cruise time window
μSC_i	The copy interval of flight leg i
e_i	The congestion coefficient of the destination station of

	flight leg i
TA	The base turnaround time
TA_i	The turnaround time in the destination station of flight leg i
F	The maximum flying time between two successive A-checks
T	The maximum number of takeoffs between two successive A-checks
D	The maximum number of flying days between two successive A-checks
C_k	The aircraft daily usage cost
C_f	The unit fuel cost
C_c	The unit CO_2 emission cost
C_l	The unit idle time cost
W_{ms}	The workforce capacity of maintenance station ms
d	The planning horizon

(1) Network structure

The AMRP-CSC is defined on a modified connection network $G = (V, A)$ that is based on the connection network established by Al-Thani et al. (2016). The node set V consists of source node o, sink node s, aircraft nodes k $(k \in K)$ and subset LN $\{11, \ldots, in, \ldots, jm, \ldots\}$ corresponding to flight leg i $(i \in L)$ copy n $(n \in N_i)$. The set of arcs A is composed of five sub-sets of arcs A^O, A^K, $A^L$and A^S. The definition of the five sub-sets is described below:

i) An arc $(o, k\ )$ $(k \in K)$ in A^O goes from the source node o to an aircraft node k;

ii) An arc $(k, in) \in A^K$ $(k \in K,\ in \in LN)$ if and only if the aircraft k is initially located at the origin station of flight leg i;

iii) An arc $(in, jm) \in A^L$ $(in, jm \in LN)$ if and only if: 1) the origin station of flight leg j is identical to the destination station of flight leg i, 2) the time interval between the departure time of flight leg j copy m (DT_j) and the arrival time of flight leg i copy n (AT_{in}) is not less than the minimum turnaround time TA_i;

iv) An arc $(jm, s\ )$ $(jm \in LN)$ in A^S connects the flight leg j copy m to a sink node s.

Clearly, the $o - s$ paths in G correspond to aircraft routes.

(2) Model formulation

To formulate the model, some additional sets and parameters are used:

σ_{in}^{-}: set of inbound arcs to flight i copy n $(in \in LN)$.

σ_{in}^{+}: set of outbound arcs from flight i copy n $(in \in LN)$.

$A^M \subseteq A^L$: set of maintenance arcs. An arc $(in, jm) \in A^M$ if and only if the origin station of flight leg j is a maintenance station and the time interval between the departure time of flight leg j copy m (DT_j) and the arrival time of flight leg i copy n (AT_{in}) is not less than the time required for A-check MT

$A^{\bar{M}} = A^L \backslash A^M$: set of non-maintenance arcs.

LN^M: set of maintenance nodes. A node $in \in LN^M$ $(in \in LN)$ if and only if flight leg i copy n has at least one inbound maintenance arc.

$LN^{ms} \in LN^M$: a node $in \in LN^{ms}$ if and only the origin station of flight leg i copy n is identical to the maintenance station ms $(ms \in MS)$.

h_{injm}: binary constant that takes the value of 1if and only if the departure time of flight leg i copy n occur one day before the departure time of flight leg j copy m.

The decision variables are defined as below:

x_{ok} =1, if arc $(o, k) \in A^O$is chosen
=0, otherwise

x_{kin} =1, if arc $(k, in) \in A^K$ is chosen
=0, otherwise

x_{injm} =1, if arc $(in, jm) \in A^L$ is chosen
=0, otherwise

x_{jms} =1, if arc $(jm, s) \in A^S$ is chosen
=0, otherwise

y_{in} =1, if a A-check is arranged before operating the flight leg i copy n, $in \in LN^M$

=0, otherwise

u_{in} a continuous variable denoting the accumulative flying time after operating flight leg i copy n, since the last A-check, $in \in LN$

v_{in} a continuous variable denoting the accumulative number of take-offs after operating flight leg i copy n, since the last A-check, $in \in LN$

w_{in} a continuous variable denoting the accumulative number of days after operating flight leg i copy n, since the last A-check, $in \in LN$

Based on the aforementioned notations, the mathematical model of the presented problem is described below:

$$min(C_1 + C_2 + C_3) \tag{5.4}$$

$$C_1 = \sum_{(o,k)\in A^O} x_{ok} \times C_k \times d \tag{5.5}$$

$$\begin{aligned} C_2 = \Bigg[\sum_{(k,in)\in A^K} x_{kin} \times DT_i + \sum_{(in,jm)\in A^{\overline{M}}} x_{injm} \times (DT_j - AT_{in} - TA_i) \\ + \sum_{(in,jm)\in A^M} x_{injm} \times (DT_j - AT_{in} - MT) \\ + \sum_{(jm,s)\in A^S} x_{jms} \times (24 \times D \times 60 - AT_{jm}) \Bigg] \times C_l \end{aligned} \tag{5.6}$$

$$\begin{aligned} C_3 = \Bigg[\sum_{(k,in)\in A^K} Fuel_{in} \times x_{kin} + \sum_{(in,jm)\in A^L} Fuel_{jm} \times x_{injm} \Bigg] \times (\varepsilon C_c \\ + C_f) \end{aligned} \tag{5.7}$$

$$\begin{aligned} Fuel_{in} = \lambda_1 \, (1/CT_{in}) + \lambda_2 \, (1/CT_{in})^2 + \lambda_3 (CT_{in})^3 + \\ \lambda_4 (CT_{in})^2, in \in N_i \end{aligned} \tag{5.8}$$

$$x_{sk} = \sum_{in \in LN} x_{kin} \, , k \in K \tag{5.9}$$

$$\sum_{(in,jm)\in\sigma_{jm}^{-}} x_{injm} + \sum_{(k,jm)\in\sigma_{jm}^{-}} x_{kjm} = \sum_{(jm,in)\in\sigma_{jm}^{+}} x_{jmin} + \sum_{(jm,s)\in\sigma_{jm}^{+}} x_{jms}\,, jm \in LN \tag{5.10}$$

$$\sum_{m\in N_j}\sum_{(in,jm)\in\sigma_{jm}^{-}} x_{injm} + \sum_{m\in N_j}\sum_{(k,jm)\in\sigma_{jm}^{-}} x_{kjm} = 1, j \in L \tag{5.11}$$

$$y_{jm} \le \sum_{(in,jm)\in A^M} x_{injm}\,, jm \in LN^M \tag{5.12}$$

$$u_{jm} \le u_{in} + FT_{jm} + (1 - x_{injm}) \times (T - FT_{in} - FT_{jm}), (in,jm) \in A^L \tag{5.13}$$

$$u_{jm} \ge u_{in} + FT_{jm} - (1 - x_{injm}) \times F, (in,jm) \in A^{\overline{M}} \tag{5.14}$$

$$u_{jm} \ge u_{in} + FT_{jm} - (1 - x_{injm}) \times F - y_{jm} \times F\,, (in,jm) \in A^M \tag{5.15}$$

$$u_{jm} \le F - y_{jm} \times (F - FT_{jm}), jm \in LN^M \tag{5.16}$$

$$u_{jm} \le FT_{jm} + (1 - x_{kjm}) \times (F - FT_{jm}), (k,jm) \in A^K \tag{5.17}$$

$$v_{jm} \le v_{in} + 1 + (1 - x_{injm}) \times (T - 2), (in,jm) \in A^L \tag{5.18}$$

$$v_{jm} \ge v_{in} + 1 - (1 - x_{injm}) \times T, (in,jm) \in A^{\overline{M}} \tag{5.19}$$

$$v_{jm} \ge v_{in} + 1 - (1 - x_{injm}) \times T - y_{jm} \times T\,, (in,jm) \in A^M \tag{5.20}$$

$$v_{jm} \le T - y_{jm} \times (T - 1), jm \in LN^M \tag{5.21}$$

$$v_{jm} \le 1 + (1 - x_{kjm}) \times (T - 1), (k,jm) \in A^K \tag{5.22}$$

$$w_{jm} \le w_{in} + h_{injm} + (1 - x_{injm}) \times (D - h_{injm} - 1), (in,jm) \in A^L \tag{5.23}$$

$$w_{jm} \ge w_{in} + h_{injm} - (1 - x_{injm}) \times D, (in,jm) \in A^{\overline{M}} \tag{5.24}$$

$$w_{jm} \ge w_{in} + h_{injm} - (1 - x_{injm}) \times D - y_{jm} \times D\,, (in,jm) \in A^M \tag{5.25}$$

$$w_{jm} \le D - y_{jm} \times (D - 1), jm \in LN^M \tag{5.26}$$

$$w_{jm} \le 1 + (1 - x_{kjm}) \times (D - 1), (k,jm) \in A^K \tag{5.27}$$

$$\sum_{in\in LN^{ms}} y_{in} \le W_{ms} \qquad ms \in MS \tag{5.28}$$

$$FT_{jm} \le u_{jm} \le F, jm \in LN \tag{5.29}$$

$$1 \le v_{jm} \le T, jm \in LN \tag{5.30}$$

$$1 \leq w_{jm} \leq D, jm \in LN \tag{5.31}$$

$$x_{ok} \in \{0,1\}, (o,k) \in A^O \tag{5.32}$$

$$x_{kin} \in \{0,1\}, (k,in) \in A^K \tag{5.33}$$

$$x_{injm} \in \{0,1\}, (in,jm) \in A^L \tag{5.34}$$

$$x_{jms} \in \{0,1\}, (jm,s) \in A^S \tag{5.35}$$

$$y_{in} \in \{0,1\}, in \in LN^M \tag{5.36}$$

The objective (5.4) is to minimize the sum of aircraft usage costs, idle time costs and fuel-burn related costs, which are calculated by equation (5.5), (5.6) and (5.7) respectively. In equation (5.6), the turnaround time is estimated based on the airport congestion coefficients, calculated by $TA_i = e_i \times TA$. This is reasonable since it takes longer time for an aircraft to visit a congested airport. To calculate the $Fuel_{in}$ in equation (5.7), we ignore the fuel consumption change in the non-cruise stage and only model the fuel consumption in the cruise stage (Duran et al., 2015). The cruise stage fuel flow model which is established by EUROCONTROL (2013) and defined in equation (5.8), is adopted. λ_1, λ_2, λ_3 and λ_4 represent the fuel consumption coefficients that are specified in the BADA user manual (EUROCONTROL, 2013). As stated in EUROCONTROL (2001), every kilogram of fuel burnt produce 3.15 kilograms of CO_2, therefore, ε in equation (5.7) is the constant of CO_2 emission that equals 3.15.

Constraints (5.9) ensures that there must be one flight leg copy following the used aircraft in the model. Constraints (5.10) is the balance constraint that ensures the circulation of an aircraft all over the connection network. Constraint (5.11) is the coverage constraint guaranteeing that exactly one copy can be selected by each flight leg and each flight leg must be flown by only one route. Constraint (5.12) ensures that the A-check is not permitted before the flight leg j copy m if the maintenance arc inbound to flight leg j copy m is not selected. Constraints (5.13) -(5.17), (5.18) -(5.22) and (5.23) -(5.27) define the accumulative flying time, the accumulative number of take-offs and the accumulative number of consecutive flying days after an aircraft operates the flight leg j copy m since the last A-check. Constraints (5.29) -(5.31) describe the three maintenance requirements including the maximum flying time, maximum number of take-offs and maximum number of consecutive flying days between two A-checks. Constraint (5.28) considers the availability of the maintenance workforce and ensures the sufficient workforce when an aircraft visits a maintenance station. The decision variables' domain are described in constraints (5.32) -(5.36).

5.2.2.2 Detailed steps of the NS procedure

To specify this phase in detail, the following notations are used. BS is defined as the incumbent best solution that consists of h aircraft routes $r_1, \ldots\ldots, r_h$. The reduced instance represented by $RS(RS \subseteq BS)$ is built by extracting a specific number of aircraft routes from the BS. This specific number is denoted by nr. nr_{min} and nr_{max} is the minimum and maximum possible value of nr, respectively. The NS approach is described below:

Step 1: Starting from the$\boldsymbol{BS}$. Set the iteration number ϑ to be 1 and $nr = nr_{min}$.

Step 2: Constructing the $\boldsymbol{RS}$. The strategy for constructing the RS is specified below:

a) Calculate the average number of flight legs per route in BS, which is denoted by fl_{ave}, by dividing the total number of flight legs FN by the number of used aircraft h.

b) Based on the value of fl_{ave}, the set of routes in BS is divided into two subsets κ and $\lambda\kappa$. κ contains the routes in which the number of flight legs is larger than value of fl_{ave} while the $\lambda\kappa$ is its complementary set. The number of routes in κ is μ_1 and in $\lambda\kappa$ is μ_2.

c) if $nr - 1 <= \mu_2$, randomly select $nr - 1$ routes in $\lambda\kappa$ and one route in κ to generate the RS, otherwise randomly select μ_2 routes in $\lambda\kappa$ and $nr - \mu_2$ routes in κ to generate the RS. Note that when generating the RS, besides the flight legs with the specific cruise times in the selecting routes, the same flight legs with different cruise times should also be added to the RS.

Step 3: Optimizing the $\boldsymbol{RS}$. Formulate the RS as a MILP model given in (5.4)-(5.36) and optimally solve the model.

Step 4: Updating the $\boldsymbol{BS}$. If a better solution is derived, then i) update the BS by substituting the old routes with the obtained solution, ii) set $\vartheta = \vartheta + 1$, iii) go to step 5, otherwise go to step 5**.**

Step 5: if $\vartheta <= \vartheta_{max}$, go to step 2, otherwise go to step 6.

Step 6: if $nr < nr_{max}$, set $nr = nr + 1$, $\vartheta = 1$ and proceed to step 2, otherwise output the BS and end.

5.2.3 Overview of the matheuristic approach

How to design a hybrid solution approach by aforementioned components that can better balance the trade-off between computational efficiency and improvement potential is a great challenge. Two strategies are considered. The first strategy is to iteratively run the exact approach during the IACO algorithm, while the second strategy is to execute it afterwards and only once.

Two strategies are tested on many instances with distinct characteristics. In the case of the NS procedure, the first strategy yields a negligible improvement in solution quality at a cost of a substantial increase in computation time, whereas the second strategy is simple but more powerful. This, however, is not the case with the SP procedure. We notice a remarkable improvement in solution quality when the first strategy is applied in the medium and large-size problems, though the computation time is much longer.

As a result, two schemes, namely ACO-SP-NS-a and ACO-SP-NS-b, are designed for the proposed matheuristic. The idea of the ACO-SP-NS-a is quite straightforward: it combines the IACO algorithm, the SP process, and the NS procedure into a sequential scheme. In the case of the ACO-SP-NS-b, despite the same application strategy of the NS procedure, the SP procedure is iteratively called after the number of performed iterations reaches a parameter PI in IACO algorithm.

We make two observations to enhance the efficiency of the proposed matheuristic. To begin, it is known that Mix Integer Programming (MIP) solver generally adopts a branch-and-bound or a branch-and-cut procedure. To increase the efficiency of the MIP solver, we improve its cutoff ability by utilizing the current best solution as the initial upper bound. Second, when the number of solutions in the pool hits a certain threshold, we update the pool by substituting the poorest solution for a better one. To avoid the inefficient updating, in ACO-SP-NS-b, if the SP procedure fails to produce a better solution based on the updated pool, this updating will be cleared.

The outlines of ACO-SP-NS-a and ACO-SP-NS-b are presented in algorithm 1 and algorithm 2, respectively. Due to the quite straightforward idea of the ACO-SP-NS-a, we shall skip the interpretation of the ACO-SP-NS-a, and instead focus on the ACO-SP-NS-b. The main IACO-SP cycle (lines 4-24) terminates after NI iterations. Each iteration will generate a new solution using the state transition mechanism specified in Section 4.1, which will be utilized to update the existing best solution if it achieves a better objective value (lines 5-8). This new solution is then added to Ω if the total number of solutions in Ω is less than the predetermined value pn (lines 9-10). Otherwise, the Ω is updated by replacing the worse solution with the new solution if the new solution achieves a better objective value (lines 11-12). The pn denotes the maximum allowable number of solutions that can be entered in Ω , which is set to ensure the tractability of the SP model. When the number of performed iterations reaches a specific value PI, the SP procedure defined in section 4.2 is iteratively executed (lines 13-15). In this manner, not only a better initial upper bound can be provided, but also the long computational time can be avoided. It worth noting that we avoid the duplicates before the execution of the SP procedure (line 14). After the execution of the SP procedure, the current best solution and Ω are updated if a better solution can be derived (lines 16-18). Otherwise, the updating of Ω done in this iteration will be cancelled (lines 19-21). Finally, the pheromone trail is updated using the incumbent best solution by adopting the pheromone updating mechanism described in section 4.1 (line 23). A final improvement of the solution is achieved by the NS procedure described in section 4.3 in lines 26-32.

Algorithm 1: IACO-SP-NS-a

```
1.  Input: a set of flight legs $L$; a set of maintenance stations $MS$; a set of aircraft $K$; number of iterations $NI$ and $\vartheta_{max}$; adjustable parameters $nr_{min}$, $nr_{max}$
2.  Output: $BS$
3.  Initialization: $\Omega \leftarrow \Phi$; $BS \leftarrow \Phi$; $obj(BS) \leftarrow \infty$
4.  for $j$ = 1 to $NI$ do
5.      Construct a complete solution (a set of aircraft maintenance routes) $S_j$ by using the state transition mechanism;
6.      if $obj(S_j)< obj(BS)$ then
7.          $BS \leftarrow S_j$;
8.      end if
9.      Update the pheromone trail by using the global pheromone trail update rule
10.     if $j <= pn$ do
11.         Add the routes of $S_j$ to $\Omega$;
12.     elseif $j > pn$ and $obj(S_j) < max(obj(S))$ (where $max(obj(S))$ represents the poorest solution in $\Omega$ which has the maximum objective value)
13.         Delete the routes of poorest solution $S$ in $\Omega$;
14.         Add the routes of $S_j$ to $\Omega$;
15.     end if
16. end for
17. Delete the duplicated routes in $\Omega$
18. $BS :=$ Solve the SP model over the set of routes stored in $\Omega$
19. for $nr = nr_{min}: nr_{max}$
20.     for $\vartheta = 1: \vartheta_{max}$
21.         Constructing the $RS$ using the routes of $BS$
22.         Optimizing the $RS$
```

```
        Updating the $BS$
    end for
end for
return $BS$
```

Algorithm 2: IACO-SP-NS-b

```
Input: a set of flight legs $L$; a set of maintenance stations $MS$; a set of aircraft $K$; number of iterations $NI$ and $\vartheta_{max}$; adjustable parameters $nr_{min}$, $nr_{max}$
Output: $BS$
Initialization: $\Omega \leftarrow \Phi$; $BS \leftarrow \Phi$; $obj(BS) \leftarrow \infty$
for $j$ = 1 to $NI$ do
    Construct a complete solution (a set of aircraft maintenance routes) $S_j$ by using the state transition mechanism;
    if $obj(S_j)$< $obj(BS)$ then
        $BS \leftarrow S_j$;
    end if
    if $j <= pn$ do
        Add the routes of $S_j$ to $\Omega$;
    elseif $j > pn$ and $obj(S_j) < max(obj(S))$ (where $max(obj(S))$ represents the poorest solution in $\Omega$ which has the maximum objective value)
        Updating the $\Omega$: delete the routes of the poorest solution $S$ in $\Omega$ and add the routes of $S_j$ to $\Omega$;
        if $j >= PI$ do
            Delete the duplicated routes in $\Omega$;
            $S_{SP} :=$ Solve the SP model over the set of routes stored in $\Omega$;
            if $obj(S_{SP}) <= obj(BS)$ then
                    $BS \leftarrow S_{SP}$;
                    Updating the $\Omega$: delete the routes of the poorest solution $S$ in $\Omega$ and add the routes of $S_j$ to $\Omega$;
            else
```

```
                        Clear the updating of this iteration in $\Omega$;
                **end if**
            **end if**
        **end if**
        Update the pheromone trail by using the global pheromone trail update rule;
    **end for**
    **for** $nr = nr_{min}: nr_{max}$
        **for** $\vartheta = 1: \vartheta_{max}$
            Constructing the $RS$ using the routes of $BS$;
            Optimizing the $RS$;
            Updating the $BS$;
        **end for**
    **end for**
    return $BS$
```

5.3 Computational experiments

In this section, the two schemes of the matheuristic approach, namely IACO-SP-NS-a and IACO-SP-NS-b, are compared first. Then the proposed matheuristic is compared to the CPLEX solver on small-size test cases, and three meta-heuristic approaches, i.e., IACO, GA and SA, on medium and large-size test cases, so that its efficiency can be verified. For each test case, all the algorithms are independently examined for 10 times in order to mitigate the impact of the randomness. Our algorithms are implemented in MATLAB R2016a and the CPLEX 12.1is adopted to tackle the MILP model. All the experiments are carried out on a 2.60 GHz Intel® Core i7 processor and 16 GB RAM computer.

The test cases are extracted from the BTS (BTS, 2020) database. They comprise ten flight schedules with the number of flight legs ranging from 40 to 2100. In general, a schedule with fewer than 250 flight legs is considered to be the small-size case (Eltoukhy et al. 2018a; Ruan et al. 2021). Therefore, our study contains five small-size test cases (instance 1- instance 5) and five large-size test cases (instance 6-instance 10). The overview of the test cases is provided in Table 5.1. The parameters setting of the proposed matheuristic plays an important role in balancing computation time and solution quality. Preliminary experiments are performed to calibrate these parameters so as to achieve a good compromise. Their values are presented in Table 5.2. Based on the recommendation of airlines and referring to the study of Duran et al. (2015), the parameters regarding the experiment setup are presented in Table 5.3. With respect to the aircraft type A320 111, the parameters regarding the fuel consumption are presented in Table 3.2.

Table 5.1: Test instances

Test cases	Number of flight legs	Number of airports	Maintenance stations
1	40	14	7
2	83	24	10
3	120	42	16
4	150	53	18
5	222	56	18
6	410	57	21
7	613	57	21
8	941	146	41
9	1358	112	28
10	2100	174	58

Table 5.2: The value of the parameters used in the proposed matheuristic

Parameter	Value	
	Small-size problems	Large-size problems
nr_{min}	3	3
nr_{max}	3	5
PI	/	50
pn	30	20
NI	100	
ϑ_{max}	10	

Table 5.3: The value of the parameters regarding the experiment setup

Parameter	Value

Aircraft type	A320 111
$C_k(US\$/day)$	$600 \times C_l$
$C_l(US\$/minute)$	136
$C_f(US\$/kg)$	0.33
$C_c(US\$/kg)$	0.02
$F(hours)$	40
T	8
$D\ (days)$	4
$TA\ (minute)$	28
$MT\ (hours)$	6
$[SC_i(1-\gamma), SC_i]$	$[85\% \times SC_i, SC_i]$
μSC_i	$5\% \times SC_i$

5.3.1 Performance comparison between two schemes

This section compares IACO-SP-NS-a and IACO-SP-NS-b and reports the comparison results in Table 5.4. In this table, S_{best}, S_{ave}, SD and $\bar{T}$ depict the best solution, the average solution, the standard deviation, and the average computation time in seconds over the 10 executions of the proposed method. A performance indicator $\%GAP$ is introduced to represent the percentage difference between the average solutions of two schemes. It is calculated by $|100(S_{ave}^a - S_{ave}^b)/S_{ave}^b|$, where S_{ave}^a and S_{ave}^b denote the average solution of IACO-SP-NS-a and IACO-SP-NS-b, respectively. The final T_{diff} column reports the difference in computation times of two schemes.

As illustrated in Table 5.4, the performance comparisons of the two schemes varies with the test case size. Regarding the small-size cases, we don't observe any substantial difference in solution quality between the two schemes. They offer the same S_{best}, while S_{ave} and SD of IACO-SP-NS-b are slightly better than those of IACO-SP-NS-a. When it comes to the medium and large-size cases, the advantage of IACO-SP-NS-b is much more pronounced, as S_{best}, S_{ave} and SD are far much better than those of IACO-SP-NS-a. Here suggests two possible explanations for the high solution quality of IACO-SP-NS-b: (i) implementing the SP procedure for multiple times and (ii) the higher-quality solution that has been improved by the SP procedure in each iteration provides better guidance for the subsequent search in the ACO framework. Obviously, the computation time of IACO-SP-NS-b is significantly longer, as solving the SP model iteratively is much more time-consuming. This is acceptable for the medium and large-size cases due to the proportional improvement in solution quality. For example, a 148 seconds increase in computation time has improved the average solution by around 23.28% in case 6. However, for small-size cases, this is not worthwhile since the average solutions are fairly comparable with the maximum variance of 0.1329% and the average difference of 0.034%.

In sum, to offer a better compromise between solution quality and computation time, we apply the IACO-SP-NS-a to tackle small-size cases and IACO-SP-NS-b to tackle medium and large-size cases. Further experiments are performed based on this application strategy.

Table 5.4: Performance comparison between ACO-SP-NS-a and ACO-SP-NS-b

Test case	ACO-SP-NS-a				ACO-SP-NS-b				$\%GAP$	T_{diff}(s)
	S_{best}	S_{ave}	SD	$\overline{T}$(s)	S_{best}	S_{ave}	SD	$\overline{T}$(s)		
1	953956	953956	0	23.00	953956	953956	0	61.08	0	38.08
2	4741996	4741996	0	25.11	4741996	4741996	0	69.67	0	44.56
3	10987985	10988194	419	31.80	10987985	10988089	314	86.07	0.001	54.27
4	4763484	4767307	2504	39.67	4763484	4765569	1042	111.56	0.0365	71.89
5	6636761	6647974	7265	59.20	6636761	6639148	2216	150.48	0.1329	91.28
6	15575372	15649104	102300	145.25	12679469	12693806	43011	294.03	23.2814	148.78
7	21735549	23505942	938144	223.14	19608377	19866306	395007	515.70	18.3206	292.56
8	54552712	55473624	550575	275.07	48865172	50010514	592332	700.03	10.9239	424.96
9	54094574	57358755	1622029	506.40	53669896	54548068	438316	1215.83	5.1527	709.42
10	175515034	176997682	1501094	953.88	170194437	171169277	537946	2472.38	3.4051	1518.51

5.3.2 Performance on small-size cases

In this section, the MILP model presented in section 5.2.1 is directly solved by the CPLEX solver, and thereby the optimal solutions are obtained, which can be used to evaluate the accuracy of IACO-SP-NS-a. Table 5.5 compares the results obtained from the CPLEX solver and IACO-SP-NS-a. Except the same notation as in Table 5.4, S^* column denotes the optimal solutions and T column represents the computation time in seconds, which are derived by the CPLEX solver. Moreover, the performance indicator $\%GAP$ is calculated by $\%GAP = 100\ (S_{ave} - S^*)/S^*$.

According to Table 5.5, S_{best} and S_{ave} of the ACO-SP-NS-a are equal to S^* in the first two test cases. As the case size increases, S_{best} remains equal to S^* while S_{ave} slightly differs from S^* by no more than 0.17%. In reference to the SD column, we find that there is no solution variance in the first two test instances. For the remaining three test instances, the solution variance increases but is quite minimal. Thus, the accuracy and the reliability of ACO-SP-NS-a have been clearly demonstrated.

Regarding the computation time, Table 5.5 shows that the computation time of ACO-SP-NS-a are much shorter than that of the CPLEX solver except for the test case 1. ACO-SP-NS-a can solve all five test cases within 60 seconds whereas the computation time of the CPLEX solver increases dramatically with the increasing size of the test cases, even reaching 3 hours for test case 5. Figure 5.1 shows the comparison of computation time between the CPLEX solver and ACO-SP-NS-a in small-size test cases.

To summarize, ACO-SP-NS-a generates solutions of superior quality as the best solution is the same with the optimality and the average solution is very close to the optimality with a maximum deviation of less than 0.17%. Additionally, a significant reduction in computation time is achieved as compared to the CPLEX solver, particularly for larger cases.

Table 5.5: Performance comparison with CPLEX

Test cases	CPLEX		ACO-SP-NS-a				$\%GAP$
	S^*	$T(s)$	S_{best}	S_{ave}	SD	$\overline{T}(s)$	
1	953956	9.35	953956	953956	0	23.00	0
2	4741996	75.42	4741996	4741996	0	25.11	0
3	10987985	305.71	10987985	10988194	419	31.80	0.001904
4	4763484	468.29	4763484	4767307	2504	39.67	0.080246
5	6636761	10136.24	6636761	6647974	7265	59.20	0.168957

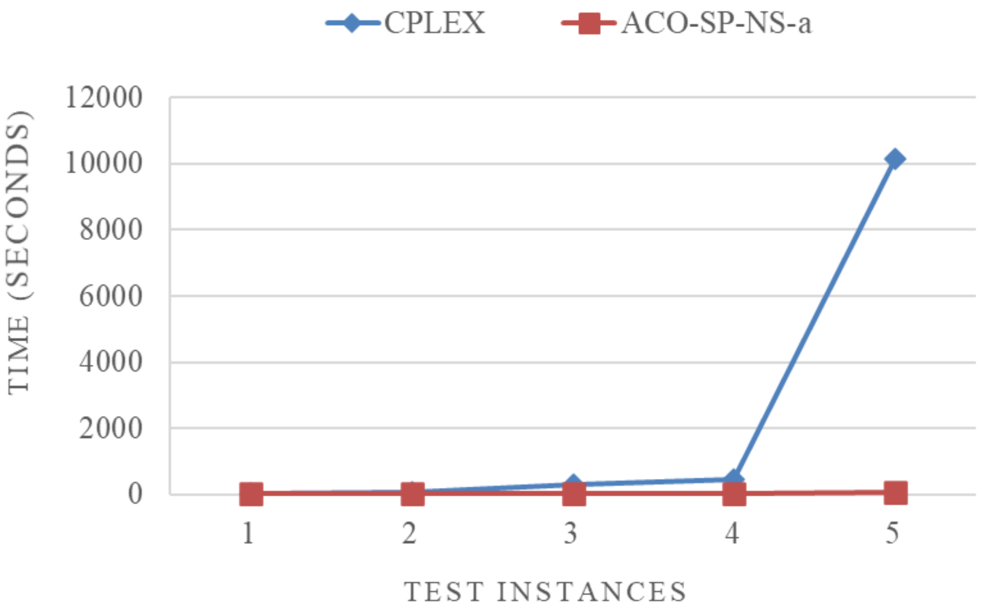


Figure 5.1: The comparison of computation time between CPLEX and ACO-SP-NS-a

5.3.3 Performance on medium and large-size cases

This section tests the proposed matheuristic on medium and large-size test cases to verify its practicality in real world conditions. As the CPLEX solver cannot obtain any feasible solutions within the time limit, results obtained from the existing promising methods are utilized as comparisons. The methods include the solely IACO algorithm described in section 4.2, Genetic Algorithms (GA) and Simulated

Annealing (SA). The user-defined parameters in the solely IACO algorithm is the same with the parameters of IACO in section 4.3.1 while that of GA and SA are set as in the Table 5.6. The parameter values of GA and SA directly follow from Eltoukhy et al. (2017b), whereas the parameter values of IACO is same with the proposed matheuristic. Table 5.7, 5.8, and 5.9 summarize the comparison results. Table 5.10 reports the percentage improvement of the solution over IACO algorithm, GA, and SA. The $\%IMR$ is calculated by $(S^C - S^b)/S^b$ where S^b and S^C represent the solution of ACO-SP-NS-b and the approach compared with ACO-SP-NS-b, respectively.

The summary results show that the ACO-SP-NS-b strongly outperforms the three methods regarding solution quality and computation time. Firstly, a remarkable improvement in solution quality can be observed. As indicated by the $\%IMR$ columns, ACO-SP-NS-b improves the average solutions over GA, SA, and IACO by an average of 32.57%, 38.93%, and 21.02%, respectively, and the best solutions over GA, SA, and IACO by an average of 31.35%, 36.41%, and 18.05%, respectively. Figure 5.2 and Figure 5.3 show the percentage improvement in average solution and best solution over GA, SA and IACO, correspondingly. Moreover, ACO-SP-NS-b achieves a noticeable reduction in computation time in comparison with the other method. The comparison of the computation time shows that ACO-SP-NS-b is even roughly a mean of 5, 1.8, and 4.2 times faster than GA, SA, and IACO, respectively. This good computation time behavior is clearly illustrated in Figure 5.4 where the computation time growth of ACO-SP-NS-b in processing increased case size is significantly slower in comparison with the other methods.

Table 5.6: The user-defined parameters setting

GA	
The population size	60
The crossover rate	0.8
The mutation rate	0.2
The maximum number of iterations	1000

SA	
The initial temperature	100
The cooling rate	0.85
The maximum number of iterations	1000

Table 5.7: Performance comparison with GA

Test cases	**ACO-SP-NS-b**			**GA**		
	S_{best}	S_{ave}	$\overline{T}(s)$	S_{best}	S_{ave}	$\overline{T}(s)$
6	12679469	12693806	294.03	20568966	21235637	1677.84
7	19608377	19866306	515.70	31561063	31561063	3218.86
8	48865172	50010514	700.03	56110089	58268558	4174.31
9	53669896	54548068	1215.83	60001075	61819963	5613.92
10	170194437	171169277	2472.38	181991071	182909461	9148.53

Table 5.8: Performance comparison with SA

Test cases	**ACO-SP-NS-b**			**SA**		
	S_{best}	S_{ave}	$\overline{T}(s)$	S_{best}	S_{ave}	$\overline{T}(s)$
6	12679469	12693806	294.03	20850209	22565191	269.58
7	19608377	19866306	515.70	30954619	31046873	690.73
8	48865172	50010514	700.03	61575522	61839229	1558.41
9	53669896	54548068	1215.83	68727883	70853261	2624.95
10	170194437	171169277	2472.38	179854580	183218982	6357.95

Table 5.9: Performance comparison with IACO

Test cases	**ACO-SP-NS-b**			**IACO**		
	S_{best}	S_{ave}	$\overline{T}(s)$	S_{best}	S_{ave}	$\overline{T}(s)$
6	12679469	12693806	294.03	16886174	17629239	300.82
7	19608377	19866306	515.70	26309212	27902182	796.92
8	48865172	50010514	700.03	56476923	58123676	4310.39
9	53669896	54548068	1215.83	56800854	59037570	5764.04
10	170194437	171169277	2472.38	172710635	173428777	18157.27

Table 5.10: Percentage improvement

Test cases	%$\boldsymbol{IMR}$					
	Average solution			**Best solution**		
	GA	**SA**	**IACO**	**GA**	**SA**	**IACO**
6	67.29%	77.77%	38.88%	62.22%	64.44%	33.18%
7	58.87%	56.28%	40.45%	60.96%	57.86%	34.17%
8	16.51%	23.65%	16.22%	14.83%	26.01%	15.58%
9	13.33%	29.89%	8.23%	11.80%	28.06%	5.83%
10	6.86%	7.04%	1.32%	6.93%	5.68%	1.48%

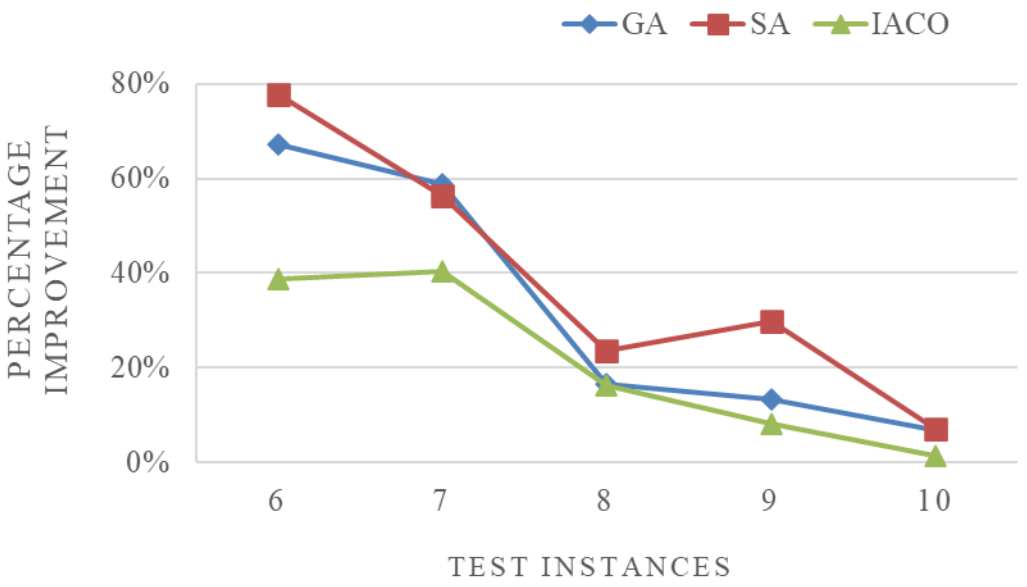


Figure 5.2: The percentage improvement in average solution over GA, SA and IACO

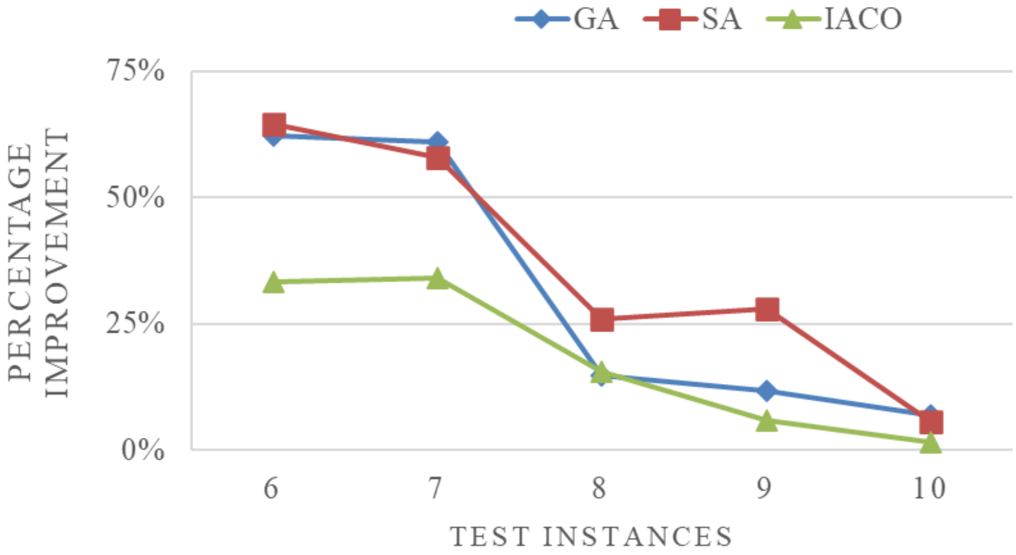


Figure 5.3: The percentage improvement in best solution over GA, SA and IACO

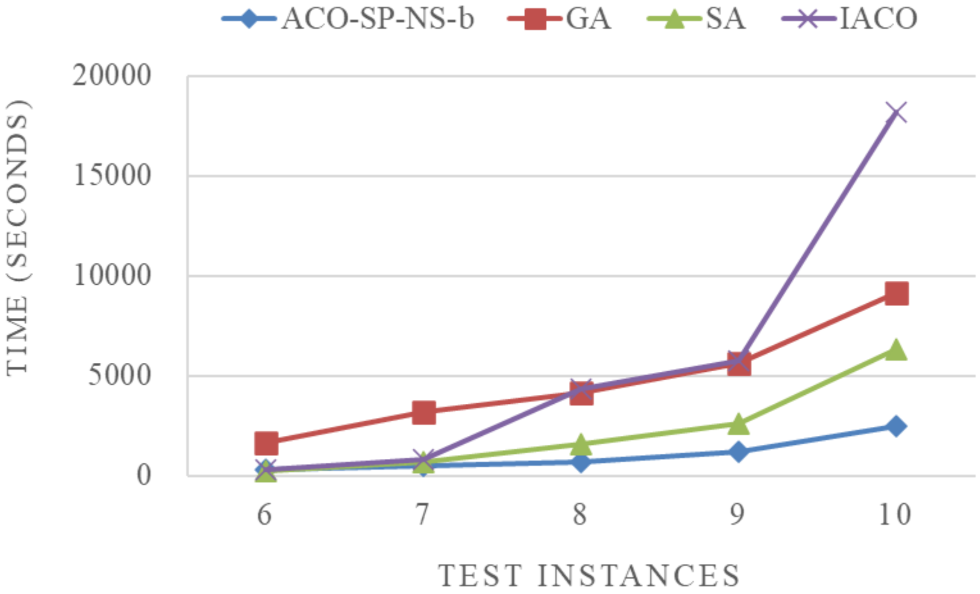


Figure 5.4: The comparison of computation times between ACO-SP-NS-a, GA, SA and IACO

5.4 Summary

In this chapter, we propose a matheuristic approach of which the critical feature is the interaction between the meta-heuristic approach and the exact method to efficiently tackle the problem. It is comprised of three main components: the IACO algorithm, the SP procedure and the NS procedure. Firstly, an IACO algorithm identifies the promising solutions to fill a pool of routes. Secondly, these routes are transformed into the columns in a SP model which is solved using the CPLEX solver to generate a superior solution. Finally, the NS procedure improves the solution by iteratively solving the reduced AMRP-CSC instances to optimality. Based on the three components, two schemes, i.e., ACO-SP-NS-a and ACO-SP-NS-b, are designed for the matheuristic. The difference in the two schemes lies in the manner that the SP procedure and the IACO algorithm are hybridized. ACO-SP-NS-a consecutively call the two procedures, while ACO-SP-NS-b embeds the SP procedure inside the iterative process of the IACO.

The proposed matheuristic is analyzed and tested using the data from BTS. Its two schemes are examined first, and the results reveal that the strategy of adopting ACO-SP-NS-a to small-size problems and ACO-SP-NS-b to medium and large-size problems yields a better compromise between the solution quality and the computational efficiency. Based on this strategy, its performance is verified by further experiments. When dealing with the small-size problems, the proposed matheuristic can identify the optimal solutions much faster than the CPLEX solver whereas the percentage differences between the average and the optimal solutions are within 0.17%. While considering medium and large-size problems, it can provide significantly higher quality solutions (an improvement percentage on average of 32.57%, 38.93% and 21.02%) within much shorter computation time (the speed-up factors on average of 5, 1.8 and 4.2 times) in comparison with the existing promising meta-heuristic approaches, i.e., GA, SA and IACO.

Chapter 6 - Robust aircraft maintenance routing problem considering disruption risk during maintenance

6.1 Introduction

Due to the prevalent and costly nature of disruption, disruption risk management lies at the core of airline operations. To mitigate the disruption risk, airlines are struggling to optimize planning decisions, i.e., schedule design, fleet assignment, aircraft routing, and crew planning (Choi et al. 2019). Robust AMRP (RAMRP) attracts more attention due to its significant impact on schedule reliability and relatively minimal impact on flight operational costs, crew costs, and passenger revenues (Lan et al. 2006). However, none of the studies investigate the disruption risk induced by aircraft maintenance. As a typical case, the annual costs of flight delays directly caused by maintenance reached over $20 million dollars for a large airline (Cook et al. 2004). Moreover, it is found that the proportion of flight delays due to maintenance was up to 12.83%, and maintenance is one of the most crucial factors leading to long-lasting flight delays, by analyzing the flight data of European airlines from 2008 to 2014 (Zámková et al. 2017). Therefore, the maintenance disruption cannot be neglected given both its severity and frequency of occurrence. To alleviate its impact, it is necessary and beneficial to model maintenance checks in a more realistic way in AMRP, to be specific, considering different task packages for an A-check program and variations in check duration for securing the reliability of the schedule.

In this sense, the crucial issue lies in how to accurately capture the impact of check duration variability on aircraft maintenance routing so as to develop response strategies to mitigate this impact. Disruption risk assessment is a good strategy; however, it is insufficient to adopt the traditional risk assessment for aircraft delays during maintenance. On one hand, it is difficult to obtain the complete and precise required information to support the traditional risk assessment due to the complexity of aircraft systems. In addition, the risk resources that may function alone or in concert to cause aircraft delays during maintenance are associated with great uncertainties. For instance, unscheduled maintenance work, as a critical risk resource, which accounts for 40–60% of maintenance requirements (Rosales et al. 2014), has no standardized tasks and is only completely known during maintenance execution. On the other hand, for each task package, due to the specific maintenance tasks involved, the probability of occurrence and severity of the same risk resource are varying, which eventually results in a distinct risk level. However, one of the significant shortcomings of traditional risk assessment is that different combinations of severity and frequency may produce an identical risk value. Consequently, it is even harder to differentiate the risk levels of distinct packages using the conventional risk evaluation. In such a situation, a fuzzy logic that can work with complex, imprecise, and uncertainty can be used.

Motivated by the above observations, we propose a novel RAMRP model with the aim of minimizing the impact of fluctuations in maintenance check duration on aircraft maintenance routing. In the proposed model, in order to match the actual scenario, we take into account several task packages for an A-check program, where each package contains certain maintenance tasks and takes a particular duration to complete. After that, a robustness improvement strategy that incorporates the assessment of delay risks is developed. To be specific, for each package, the proposed fuzzy risk assessment model precisely quantifies the delay risks caused by check duration variability. Based on the results of risk assessment, schedule robustness is achieved by more accurate task package buffer allocation during route construction. Finally, the corresponding robustness measure, namely a risk score, is proposed to quantify the total delay risks. An effective matheuristic method that consists of three primary steps, the ant colony optimization (ACO) algorithm, the set partitioning (SP) procedure, and the neighborhood search (NS) procedure, is used to solve the proposed RAMRP model.

The remaining parts of this chapter is arranged as follows. In section 6.2 we briefly review some studies regarding the RAMRP. The detailed problem description is given in section 6.3. Section 6.4 describes the fuzzy risk assessment model. The mathematical model is formulated in section 6.5. The solution method including three main components and their hybridization schemes is presented in section 6.6. The computational experiments are reported in section 6.7. Finally, we conclude with suggestions for further research in section 6.8.

6.2 Problem description

Robust AMRP considering disruption risk during maintenance is solved at the planning stage to construct a set of aircraft maintenance routes that are resilient to disruptions caused by aircraft maintenance. An aircraft maintenance route consists of a sequence of flight legs and appropriate arrangements of maintenance checks performed to satisfy the maintenance requirements mandated by Federal Aviation Administration (FAA). A-check, B-check, and C-check are the three types of preventive maintenance checks, but only A-check is considered in the proposed problem because it is the most frequently performed and thus has a greater impact on airline operational schedules. Unlike traditional AMRP, we divide the A-check program into n different task packages, i.e., A_1, A_2, ..., A_n for A-check program, each with designated maintenance tasks and a distinct duration, in order to model A-checks in a more realistic manner. A_1 has the same task package as A_{n+1}, and A_2 has the same task package as A_{n+2}, and so forth. Task packages A_1 to A_n are executed sequentially for each aircraft, and the accumulated flying time between operations of two identical task packages, such as A_1 and A_{n+1}, must be less than or equal to the maximum allowable flying time. The working times and the workforce capacity of maintenance stations are considered for each task package operation to ensure smooth operation.

In actual operation, even though each task package is assigned a specific time slot, the real execution time varies significantly due to various maintenance uncertainties, such as non-routine tasks and uncontrolled spares. The discrepancy between the planned duration and the actual duration may result in costly recovery actions, delays, or even cancellations. As a result, our study considers the check duration variability during the construction of maintenance routes. To be more specific, we accurately quantify the delay risk caused by check duration variability for each task package. The fuzzy risk assessment method is used because it can handle incomplete and imprecise information encountered in risk assessment. Subsequently, , we add extra buffer time to the task package with a higher risk level to mitigate the impact of check duration variability in real-world operation. Increasing buffer time, on the other hand, will inevitably result in lower aircraft utilization. In figure 6.1, for example, if the task package A_2 has a high delay risk level, even if aircraft's accumulated flying time reaches allowable maximum flying time after flying f_{k+2}, it is more likely to be performed between the flight connection of f_k and f_{k+1} rather than the flight connection of f_{k+2} and f_{k+3}. This is due to the much larger time interval between the arrival time of f_k and the departure time of f_{k+1}. Meanwhile, more unutilized flying time will be incurred. Figure 6.2 depicts another scenario where an extra aircraft may be required in order to allocate more buffer time to the task package A_2. As a result, to avoid a significant reduction in aircraft utilization, upper bounds for unutilized flying time and the number of used aircraft will be set in route construction. Finally, the total risk score is proposed to as a way to represent total delay risks and evaluate schedule robustness. It is calculated by adding the risk of delay for each completed task package. By constructing the maintenance routes with the minimal total risk score, our study realizes the objective of mitigating the disruption risk resulting from aircraft maintenance.

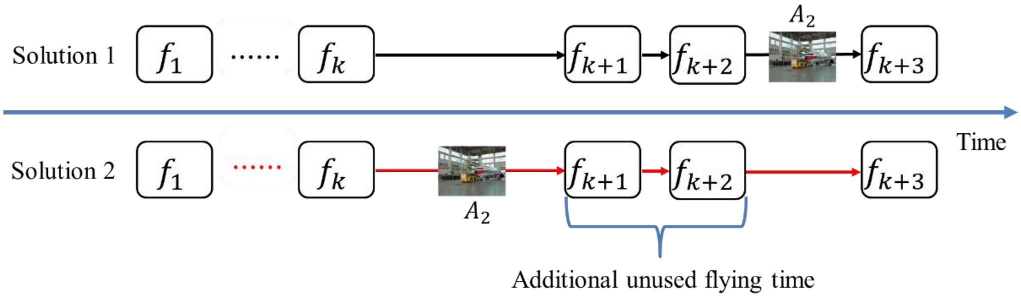


Figure 6.1: An example of additional unused flying time required

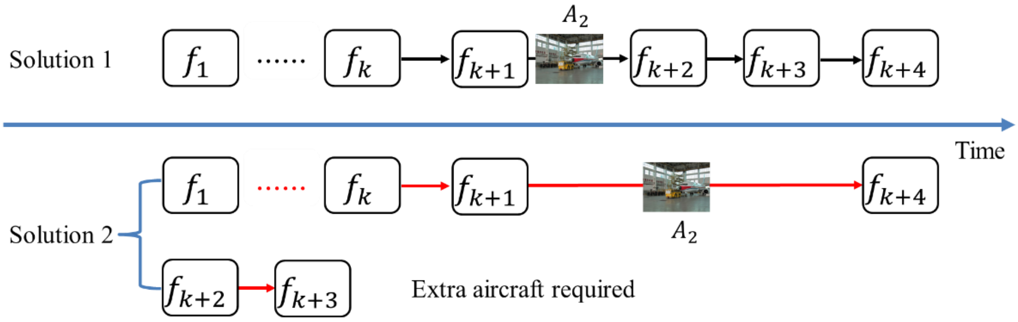


Figure 6.2: An example of extra aircraft required

6.3 Fuzzy risk assessment

Risk assessment is an analytical procedure to measure the magnitudes of an undesirable event by multiplying the likelihood of occurrences and severity of consequences. However, it is difficult to precisely obtain exact values for likelihood and severity scales, posing challenge on risk assessment. This situation would be exacerbated in risk assessment of aircraft delays during maintenance. On one hand, the risk resources that may function alone or in concert to cause aircraft delays during maintenance are associated with great uncertainties. For example, unscheduled maintenance work, as a critical risk resource, which accounts for 40–60% of maintenance requirements (Rosales et al., 2014), has no standardized tasks and is only completely known during maintenance execution. On the other hand, it is almost impossible to collect all the maintenance activities data due to the complexity of aircraft systems. However, accurate risk assessment can help airlines obtain the best cost-effective maintenance routing decision since either an underestimation or an overestimation of the delay risk will do more harm than good to airlines. Therefore, our study incorporates the fuzzy logic approach into risk assessment of aircraft delays during maintenance to tackle the incomplete information and imprecise data. The proposed fuzzy risk assessment model (FRAM) consists of four typical components:(1) risk resources identification, (2) fuzzification, (3) fuzzy inference system, and (4) defuzzification (Markowski and Mannan 2008; Skorupski 2016; Zhang et al. 2017), as shown in Figure 6.3.

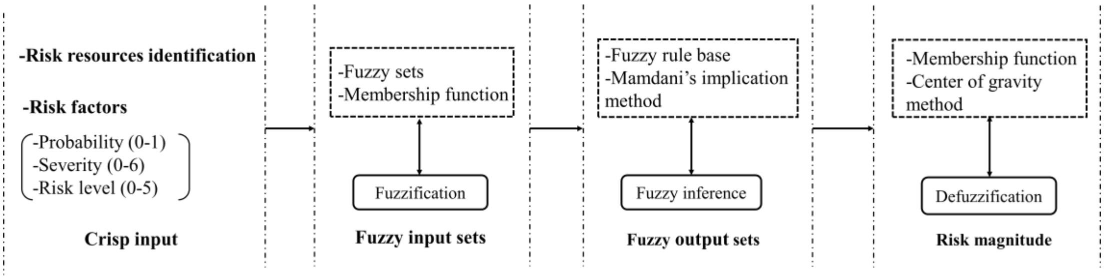


Figure 6.3: The structure of fuzzy risk assessment model

6.3.1 Risk resources identification

The identification of risk resources involves an investigation into all the potential direct causes of aircraft delays during maintenance, including their impact and corresponding occurrence. It is of considerable importance because the process of risk analysis and response strategies may only be performed on the identified potential risk resources. Based on the maintenance reports and interviews conducted with employees of airline maintenance company, the risk resources are identified and classified into different categories which are shown in Figure 6.4.

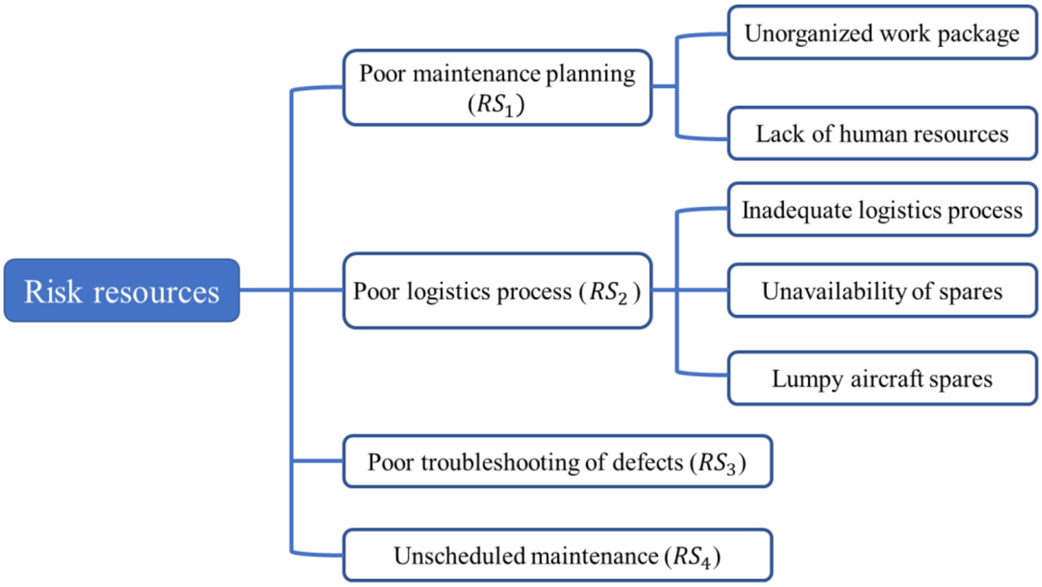

Risk resources
Poor maintenance planning (RS_1)
Unorganized work package
Lack of human resources
Poor logistics process (RS_2)
Inadequate logistics process
Unavailability of spares
Lumpy aircraft spares
Poor troubleshooting of defects (RS_3)
Unscheduled maintenance (RS_4)

Figure 6.4: The categories of risk resources

6.3.2 Fuzzification

Fuzzification that maps crisp numbers into fuzzy sets includes the determination of fuzzy sets and corresponding fuzzy membership functions. Firstly, we form fuzzy sets for the three components of the basic risk matrix, i.e., probability of a risk event, severity of a risk event, and risk magnitude. Fuzzy sets are linguistic variables whose values are either words or sentences in a natural or artificial language. These values are determined by subjective qualitative judgments based on the knowledge of specialists and the nature of the situation. In our study, the severity of delay risk takes a five-point scale: negligible, low, moderate, high, and very high. Based on the historical data of aircraft delays during maintenance, we propose a seven-point scale for the probability of delay risk: remote, unlikely, very low, low, medium, high, and very high. The probability and severity of delay risk are input linguistic variables to fuzzy inference system, while the risk level is the output linguistic variable. We take the value of risk level as very low, low, moderate, and high. Table 6.1 provides the details of fuzzy sets adopted in the delay risk assessment.

Membership functions are used to decide the membership values for each fuzzy set. In rule-based fuzzy modeling, triangular and trapezoidal membership function are the most preferred and prevalent types of membership function, which are also adopted in FRAM (Figure 6.5, 6.6 and 6.7).

Using the fuzzy sets shown in Table 6.1 and membership functions illustrated in Figure 6.5-6.7, the membership functions for measuring the probability and the severity for each delay risk resource are shown in Table 6.2.

Table 6.1: Fuzzy sets

Linguistic variables	Fuzzy set	Description range	Universe of discourse (X)
Probability (P)	Remote	$0 < P < 0.25$	$X_P \in (0,1)$
	Unlikely	$0.125 < P < 0.375$	
	Very low	$0.25 < P < 0.5$	
	Low	$0.375 < P < 0.625$	
	Medium	$0.5 < P < 0.75$	
	High	$0.625 < P < 0.875$	
	Very high	$0.75 < P < 1$	
Severity (S) (T: delay time)	Negligible $(0 < T < 15)$	$0 < S < 2$	$X_S \in (0,6)$
	Low $(15 < T < 60)$	$1 < S < 3$	
	Moderate $(60 < T < 180)$	$2 < S < 4$	
	High $(180 < T < 300)$	$3 < S < 5$	
	Very high $(300 \leq T)$	$4 < S < 6$	
Risk level (R)	Very Low	$0 < R < 2$	$X_R \in (0,5)$
	Low	$1 < R < 3$	
	Moderate	$2 < R < 4$	
	High	$3 < R < 5$	

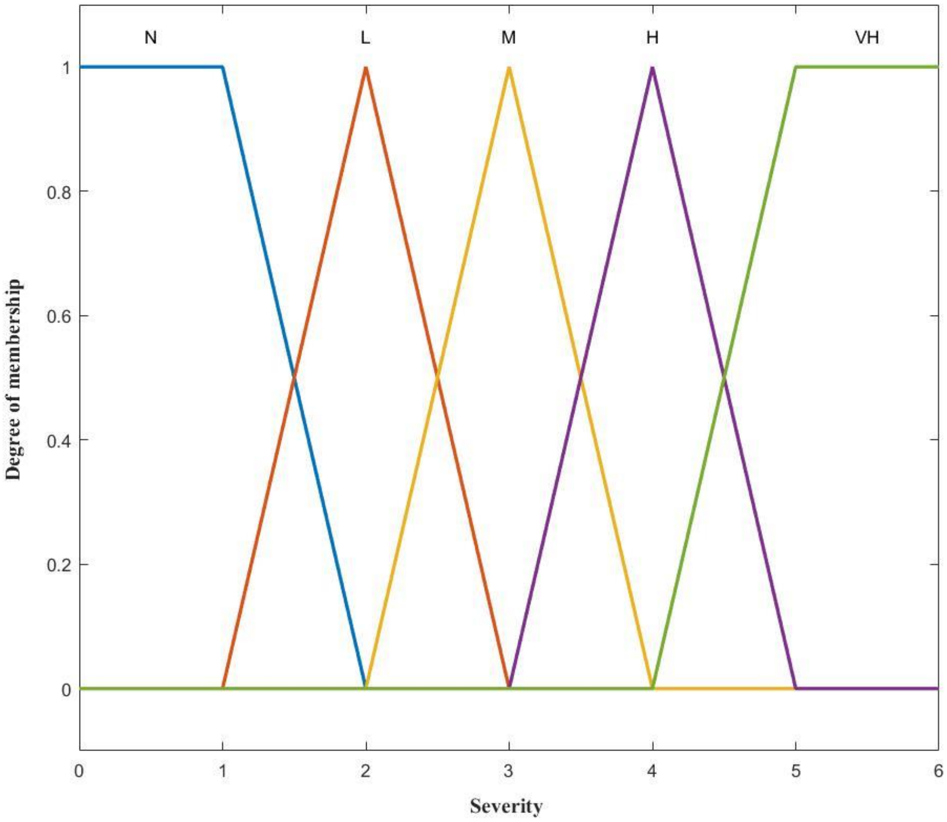


Figure 6.5: Membership function representing severity

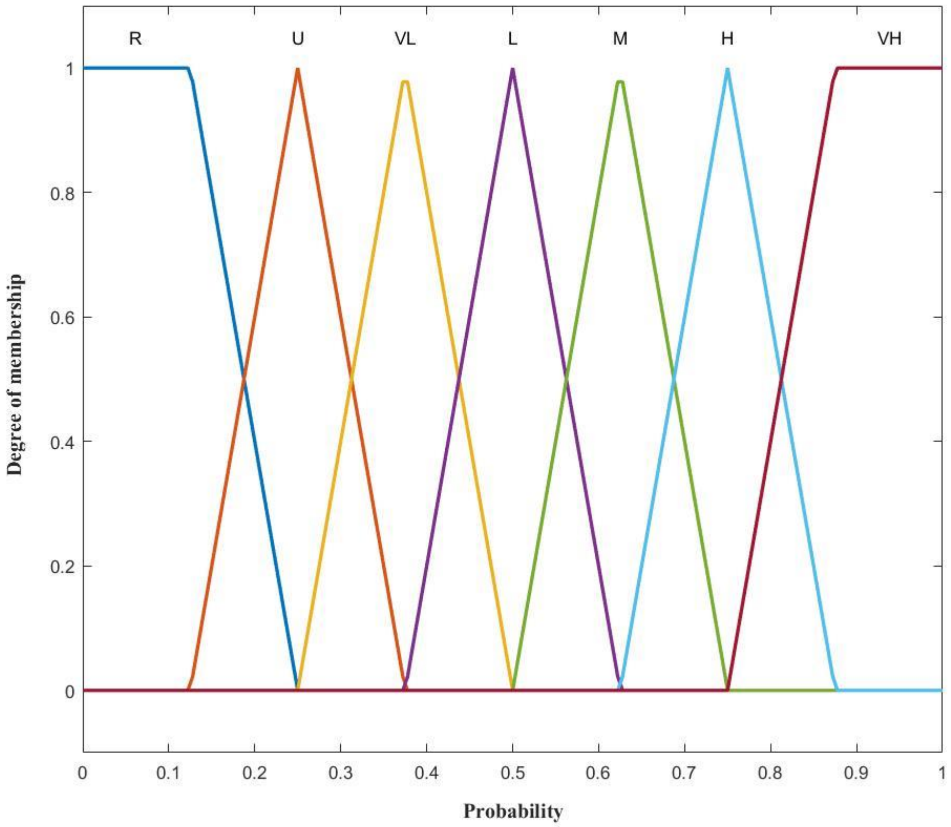


Figure 6.6: Membership function representing probability

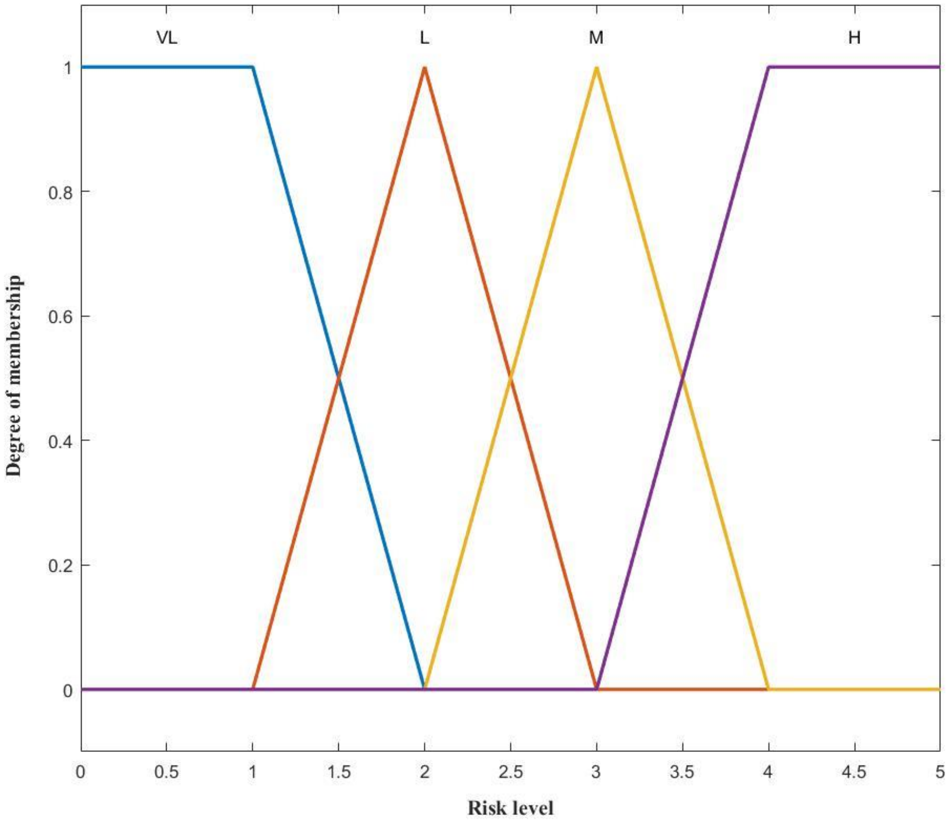


Figure 6.7: Membership function representing risk level

Table 6.2: Membership functions of risk probability and risk severity for each risk resources

Risk resources	Measure of risk probability	Measure of risk severity
RS_1	(0, 0, 1, 2)	(0.375, 0.5, 0.625)
	(1, 2, 3)	(0.625, 0.75, 0.875)
	(2, 3, 4)	(0.25, 0.375, 0.5)
RS_2	(0, 0, 1, 2)	(0.375, 0.5, 0.625)
	(2, 3, 4)	(0.625, 0.75, 0.875)
	(3, 4, 5)	(0.375, 0.5, 0.625)
	(4, 5, 6, 6)	(0125, 0.25, 0.5)
RS_3	(0, 0, 1, 2)	(0.375, 0.5, 0.625)
	(1, 2, 3)	(0.625, 0.75, 0.875)
	(2, 3, 4)	(0.25, 0.375, 0.5)
RS_4	(2, 3, 4)	(0.625, 0.75, 0.875)
	(3, 4, 5)	(0.375, 0.5, 0.625)
	(4, 5, 6, 6)	(0.125, 0.25, 0.5)

6.3.3 Fuzzy inference system

The fuzzy inference system maps two fuzzy input sets, i.e., probability and severity, into one fuzzy output set, i.e., risk level, based on fuzzy if-then rules which are obtained referring to expert knowledge. Since there are five membership functions and seven membership functions assigning to severity and probability, respectively, the number of fuzzy if-then rules is equal to 35. To be more intuitive, a risk assessment matrix with 35 risk cells can be generated to represent these fuzzy if-then rules based on the risk factors. Moreover, one limitation of the traditional risk assessment is that the different combination of probability and severity may result in the identical risk level. To avoid this shortcoming, we assign each rule a rule weight in order to distinguish the relative importance of severity and probability.

To tackle the quantitative components in fuzzy rules and aggregate the risk level of each risk source, Mamdani's implication method is adopted. It uses the minimum operator to handle the classical logic connection And in the fuzzy rule, and maximum operator to combine the results of the same fuzzy risk subset. The overview of this method is illustrated in Figure 6.8. Herein, $\overline{rp}$, $\overline{rs}$, and $\overline{rm}$ are the fuzzy subset of risk probability, risk severity, and risk level, respectively. $u_k^n(\overline{rp})$,$u_k^n(\overline{rs})$, and u_k^n $(\overline{rm})$ are the corresponding membership degrees. k is the number of rules and n is the number of risk resources.

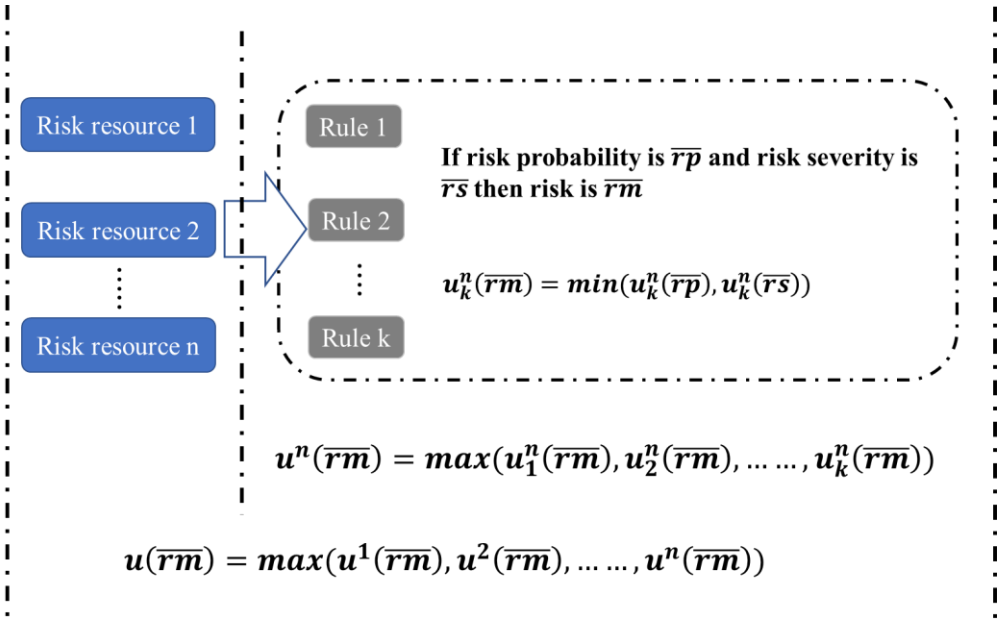

Risk resource 1
Risk resource 2
Risk resource n
Rule 1
Rule 2
Rule k
If risk probability is $\overline{rp}$ and risk severity is $\overline{rs}$ then risk is $\overline{rm}$
$u_k^n(\overline{rm}) = min(u_k^n(\overline{rp}), u_k^n(\overline{rs}))$
$u^n(\overline{rm}) = max(u_1^n(\overline{rm}), u_2^n(\overline{rm}), \ldots\ldots, u_k^n(\overline{rm}))$
$u(\overline{rm}) = max(u^1(\overline{rm}), u^2(\overline{rm}), \ldots\ldots, u^n(\overline{rm}))$

Figure 6.8: The overview of the Mamdani's implication method

6.3.4 Defuzzification

The defuzzification step, based on the fuzzy set of risk level, i.e., R , converts the fuzzy conclusion set, i.e., the pair of $\{x_R, u_{\bar{R}}(x_R)\ |x_R \in X_R\}$, into a numerical value, i.e., a risk score $x_R{}^*$, using the centroid average method, as expressed in the formula below.

$$x_R{}^* = \sum\nolimits_{\bar{R}\in R} x_R u_{\bar{R}}(x_R) \Big/ \sum\nolimits_{\bar{R}\in R} u_{\bar{R}}(x_R)$$

where x_R is an element of the universe set which the fuzzy set of risk level R is defined on, $\bar{R}$ is a member in R, and $u_{\bar{R}}(x_R)$ represents the membership degree of x_R in $\bar{R}$.

6.4 Model formulation for RAMRP considering disruption risk during maintenance

In this section, we first create the connection network by which the problem is modeled, and then formulate the mathematical model. The notations used throughout this work are defined as following:

$i, j \in F$	Set of flight legs
$k \in K$	Set of aircraft
$m \in M$	Set of maintenance stations
$a \in A$	Set of airports
$\{o, s\}$	Source node o and sink node s
$v \in \{1, \dots, V_k\}$	Number of maintenance checks that should be performed on an aircraft k
$n \in \{1, \dots, N\}$	Number of task packages for an A-check program
IT_k^n	The initial remaining time of the designated area of aircraft k for which the task package n is responsible
DT_i	The departure time of flight leg i
AT_i	The arrival time of flight leg i
FT_i	The flying time of flight leg i
MT_n	The time required to complete the task package n
D_{ia}	$D_{ia} = 1$ if the destination airport of flight leg i is identical to the airport a and $D_{ia} = 0$ otherwise.
O_{ia}	$O_{ia} = 1$ if the origin airport of flight leg i is identical to the airport a and $O_{ia} = 0$ otherwise.
TAT_k	Turn-around time for the aircraft k.
F_{max}	Maximum allowable flying time between two consecutive identical maintenance task packages.
Mb_{ma}	$Mb_{ma} = 1$ if the location of the maintenance station m is identical to the airport a and $Mb_{ma} = 0$ otherwise.
UMT	Upper bound of unutilized flying time each maintenance check.
WF_m	Workforce capacity of the maintenance station m.
ET_m	Closing time of the maintenance station m.
OT_m	Opening time of the maintenance station m.
C	Big number
RL_{imjkv}	The risk magnitude of aircraft delays when an aircraft

k performs the maintenance check number v at maintenance station m between the operations of flight leg i and flight leg j

6.4.1 Network structure

The problem is modeled by a connection network, $G = (V, E)$, which is constructed as follows. The node set V contains three disjoint subsets: (i) $\{o, s\}$ representing dummy source node and sink node that are incorporated for starting and finishing the route construction for each aircraft, (ii) set of flight leg nodes $(i, j \in F)$, and (iii) set of maintenance station nodes $(m \in M)$. Correspondingly, three types of arcs are generated including: (i) ordinary arcs that are used for an aircraft to operate two flights consecutively, (ii) maintenance arcs that are employed for an aircraft to perform a maintenance check, and (iii) auxiliary arcs that are adopted to assist an aircraft in resuming flights operations. The structure of the connection network is illustrated in Figure 6.9.

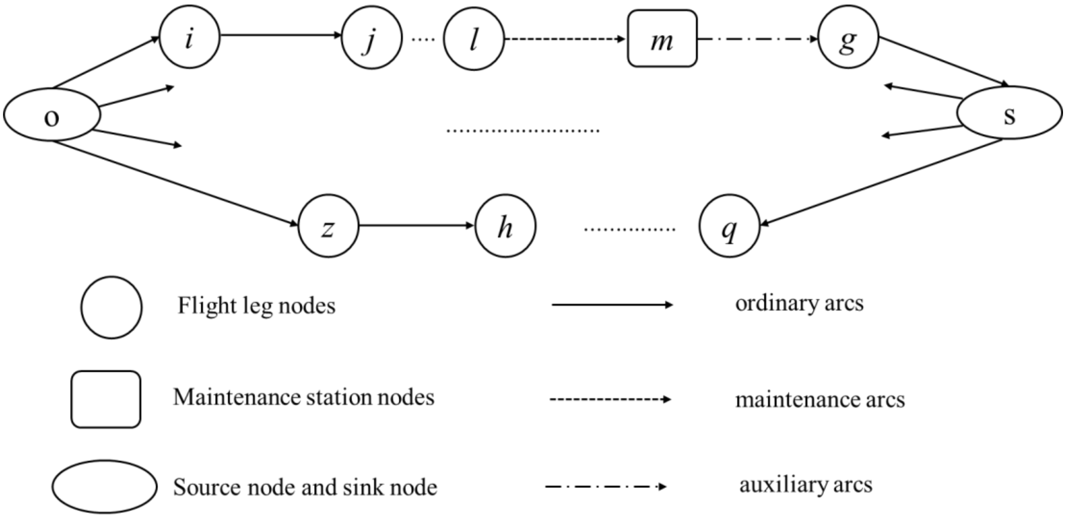


Figure 6.9: The structure of the connection network

6.4.2 Model formulation

To formulate the mathematical model, the decision variables are defined as follows:

Decision variables

$x_{ijkv} \in \{0,1\}$ $x_{ijkv} = 1$ if the aircraft k performs the flight legs i and j consecutively before undergoing the maintenance check v, and $x_{ijkv} = 0$ otherwise.

$y_{imkv} \in \{0,1\}$ $y_{imkv} = 1$ if the aircraft k performs the flight leg i and undergoes the maintenance check$v$ at the maintenance station m consecutively, and $y_{imkv} = 0$ otherwise.

$z_{mjkv} \in \{0,1\}$ $z_{mjkv} = 1$ if the aircraft k undergoes the maintenance check v at the maintenance station m and then performs the flight leg j, and $z_{mjkv} = 0$ otherwise.

$RTA_{mkv}>0$ The time when the aircraft k finishes maintenance check number v at the maintenance station m.

$UT_{mkv} \geq 0$ Unutilized flying time beyond the pre-determined upper bound when the aircraft k performs maintenance check number v at the maintenance station m.

$L \geq 0$ Number of aircraft that exceeds the fleet size.

Based on the connection network, using the above-mentioned notations and decision variables, the mathematical model of the proposed model can be written as follows:

$$min \sum_{i\in F}\sum_{j\in F}\sum_{m\in M}\sum_{k\in K}\sum_{v\in V_k} y_{imkv} z_{mjkv} \left(RL_{imjkv}\right)_\Delta + \sum_{m\in M}\sum_{k\in K}\sum_{v\in V_k}(UT_{mkv})_\Delta + (L)_\Delta \tag{6.1}$$

$$\sum_{j\in F} x_{ojkv} = 1 \quad \forall k \in K, v = 1 \tag{6.2}$$

$$\sum_{v\in V_k}\left(\sum_{i\in F} x_{iskv} + \sum_{m\in M} z_{mskv}\right) = 1 \quad \forall k \in K \tag{6.3}$$

$$\sum_{k\in K}\sum_{v\in V_k}\left(\sum_{j\in F\cup\{s\}} x_{ijkv} + \sum_{m\in M} y_{imkv}\right) = 1 \quad \forall i \in F \tag{6.4}$$

$$\sum_{j\in F\cup\{o\}} x_{jikv} = \sum_{j\in F\cup\{s\}} x_{ijkv} + \sum_{m\in M} y_{imkv} \quad \forall i \in F,\ \forall k \in K, \forall v \in V_k \tag{6.5}$$

$$\sum_{m\in M} z_{mikv} = \sum_{j\in F\cup\{s\}} x_{ijkv+1} \quad \forall i \in F,\ \forall k \in K,\ \forall v \in V_k \tag{6.6}$$

$$\sum_{i\in F} y_{imkv} = \sum_{j\in F\cup\{s\}} z_{mjkv} \quad \forall m \in MT, \forall k \in K, \forall v \in V_k \tag{6.7}$$

$$AT_i + TAT_k - DT_j \leq C\big(1 - x_{ijkv}\big) \quad \forall i,j \in F, \forall k \in K, \forall v \in V_k \tag{6.8}$$

$$x_{ijkv} \leq \sum_{a\in A} D_{ia} O_{ja} \quad \forall i,j \in F,\ \forall k \in K,\ \forall v \in V_k \tag{6.9}$$

$$y_{imkv} \leq \sum_{a\in A} D_{ia}\, Mb_{ma} \quad \forall i \in F,\ \forall k \in K, \forall v \in V_k, \forall m \in M \tag{6.10}$$

$$z_{mjkv} \leq \sum_{a\in A} Mb_{ma}\, O_{ja} \quad \forall j \in F,\ \forall m \in M,\ \forall k \in K,\ \forall v \in V_k \tag{6.11}$$

$$RTA_{mkv} \geq \sum_{i\in F} (AT_i + MT_n) y_{imkv} \quad \forall m \in M, \forall k \in K, \forall v \in V_k, n = \begin{cases} v, & v < N \\ v - N, & v > N \end{cases} \tag{6.12}$$

$$RTA_{mkv} - DT_j \leq M\big(1 - z_{mjkv}\big) \quad \forall j \in F, \forall m \in M, \forall k \in K, \forall v \in V_k \tag{6.13}$$

$$\sum_{v\in V_k} \sum_{i\in F} FT_i x_{oikv} + \sum_{i\in F} FT_j x_{ijkv} \leq IT_k^n \quad \forall k \in K, n = 1 \tag{6.14}$$

$$\sum_{v\in V_k} \sum_{i\in F} FT_i x_{oikv} + \sum_{v=1,\dots,n-1} \sum_{j\in F} \sum_{m\in M} FT_j z_{mjkv} + \sum_{v=1,\dots,n} \sum_{i\in F} FT_j x_{ijkv} \leq IT_k^n \quad \forall k \in K, \forall n \in N \tag{6.15}$$

$$\sum_{n=1,\dots,N} \sum_{j\in F} \sum_{m\in M} FT_j z_{mjk(v+n-1)} + \sum_{n=1,\dots,N} \sum_{j\in F} \sum_{i\in F} FT_i x_{jik(v+n)} \leq F_{max} \quad \forall k \in K,\ \forall v \in V_k \tag{6.16}$$

$$OT_m - AT_i \leq C(1 - y_{imkv}) \quad \forall i \in F, \forall k \in K, \forall v \in V_k, \forall m \in M \tag{6.17}$$

$$AT_i + MT_n - ET_m \leq C(1 - y_{imkv}) \quad \forall i \in F, \forall k \in K,\ \forall v \in V_k, \forall m \in M, n = \begin{cases} v, & v < N \\ v - N, & v > N \end{cases} \tag{6.18}$$

$$\sum_{i\in F} \sum_{k\in K} \sum_{v\in V_k} y_{imkv} \leq WF_m \quad \forall m \in M \tag{6.19}$$

$$UT_{mkv} \geq FP_{max} - \Big(\sum_{i\in F} FT_i x_{oikv} + \sum_{n=1,\dots,N-1} \sum_{i\in F} FT_i z_{mikv+n} + \sum_{n=1,\dots,N} \sum_{i\in F} \sum_{j\in F} FT_j x_{ijkv+n}\Big) - UMT \quad \forall m \in M,\ \forall k \in K, v = 1 \tag{6.20}$$

$$UT_{mkv} \geq FP_{max} - \Big(\sum_{n=1,\dots,N} \sum_{j\in F} \sum_{m\in M} FT_j z_{mjk(v+n-1)} + \sum_{n=1,\dots,N} \sum_{j\in F} \sum_{i\in F} FT_i x_{jik(v+n)}\Big) - UMT \quad \forall m \in M, \forall k \in K, \forall v \in V_k \setminus \{1\} \tag{6.21}$$

$$L \geq \sum_{k\in K} \sum_{j\in F} x_{ojkv} - |K| \quad v = 1 \tag{6.22}$$

$$x_{ijkv} \in \{0,1\} \quad \forall i,j \in F,\ \forall k \in K,\ \forall v \in V_k \tag{6.23}$$

$$y_{imkv} \in \{0,1\} \quad \forall i \in F,\ \forall m \in M,\ \forall k \in K,\ \forall v \in V_k \tag{6.24}$$

$$z_{mjkv} \in \{0,1\} \quad \forall j \in F,\ \forall m \in M,\ \forall k \in K,\ \forall v \in V_k \tag{6.25}$$

$$RTA_{mkv} > 0 \quad \forall m \in M,\ \forall k \in K,\ \forall v \in V_k \tag{6.26}$$

$$UT_{mkv} \geq 0 \quad \forall m \in M,\ \forall k \in K,\ \forall v \in V_k \tag{6.27}$$

$$L \geq 0 \tag{6.28}$$

In the proposed model, the restrictions on aircraft utilization are considered as soft constraints that might be violated at a penalty. The logic behind this setting is twofold: to ensure problem feasibility, and to strongly penalize large deviations from the presetting upper bounds. Accordingly, the objective function (6.1) consists of three terms. The first term refers to the total delay risks which is calculated by summing the delay risk of each task package. Herein, the delay risk of each task package is estimated using the fuzzy risk assessment model proposed in section 4. The second term defines the penalty for the unutilized flying time beyond the specified upper bound which is calculated by constraints (6.20) and (6.21). The third term represents the penalty for the additional required aircraft which is calculated by constraint (6.22). Note that the units for three terms are different, necessitating their normalization. The subscript "Δ" denotes the normalized value. The normalization function is $w_{\Delta} = \frac{w-\tilde{w}}{\bar{w}-\tilde{w}}$, where $\tilde{w}$ is the minimum value of w, whereas $\bar{w}$ is the maximum value of w. The general ideal of (6.1) is to minimize the total delay risks while having the least violations of constraints on aircraft utilization.

Constraints (6.2) -(6.4) explain the coverage constraints. Constraints (6.2) -(6.3) ensure the start and the completion of an aircraft route. The requirement that one flight leg can be operated by exactly one aircraft is defined in constraints (6.4). The balance constraint (6.5) -(6.7) is cast to keep the movement of an aircraft throughout the connection network. It depicts two different scenarios. If an aircraft visits a flight leg copy using a non-maintenance arc, the next flight leg copy must be visited using either a non-maintenance arc or a maintenance arc. In another scenario, if an aircraft visits a flight leg copy using a maintenance arc, the next flight leg copy must be visited using a non-maintenance arc.

Constraints (6.8) -(6.9) describe the location and time issues of flight connections. For two flight leg copies, which can be executed successively by the identical aircraft, the time constraint (6.8) denotes that the departure time of the latter one must be larger than the summation of the arrival time of former one and the turnaround time, while the location constraint (6.9) indicates that the arrival station of the former one must be the same as the departure station of the latter one. However, if there is a maintenance check in between, the location constraints (6.10)- (6.11) signify that the three places, including the location of the maintenance station, the arrival station of the former one and the departure station of the latter one, must be the same, while the time constraints (6.12)-(6.13) indicates that the departure time of the latter one must be larger than the summation of the arrival time of the former one and the duration of a maintenance check. Constraints (6.14) -(6.16) define the maintenance requirement that an aircraft must receive a maintenance check before accumulating the maximum allowable flying time between two identical task packages. Moreover, a maintenance check must be performed during the maintenance station's working time, which is ensured by the constraints (6.17) -(6.18). It is worth noting that durations of different maintenance checks are varying due to different task packages involved. The constraint (6.19) ensures that there is sufficient workforce when an aircraft visits a maintenance station.

At last, we define the decision variables' domain in constraints (6.23) -(6.28).

6.5 Solution method

In this section, a matheuristic approach is proposed to solve the problem. The three main components of this approach will be outlined first, and then how to merge them to efficiently tackle the proposed problem will be described.

6.5.1 The ACO algorithm

We develop an ACO algorithm to generate substantial initial solutions. Here, the related notations will be defined first and then two main parts of the ACO will be described.

$Attr_{ij}$	The attraction of arc (i, j) for an ant
q_0	A predefined parameter ($0 < q_0 < 1$)
q	A random number uniformly distributed in the range of 0 to 1
ρ	The pheromone evaporation rate ($0 < \rho < 1$)
τ_{ij}	The amount of pheromone on arc (i, j)
η_{ij}	The heuristic information on arc (i, j)
α, β	The parameter to control the relative importance of τ_{ij}, η_{ij},
$\Delta\tau_{ij}^{best}$	The pheromone increments on arc (i, j) of the incumbent best-found solution
N_i	Set of feasible nodes that can be covered right behind the node i

(1) Routes construction

Initially, an ant is positioned on the source node o of the connection network G to starts routes construction by selecting an aircraft $k \in K$. The node i with the earliest departure time will be first covered and then the following state transition mechanism presented in equation (6.28) will be used to select the next node j. Once visiting a node j, the flight leg j will be registered and cannot be selected in the future. If there is no node to be chosen due to various constraints, the ant will end the route in the sink node s. After that, the ant starts again at the source node o and repeat this process until all the flight legs are covered.

$$f(x) = \begin{cases} arg\ \_max_{j\in N_i}\{Attr_{ij}\}, if\ q \le q_0 \\ p_{ij}\,, \qquad\qquad\qquad if\ q > q_0 \end{cases} \tag{6.29}$$

When q is less than the predefined parameter q_0, the ant will choose the node j with the biggest $Attr_{ij}$ as the next node. Otherwise, the ant will randomly select the next node according to the probability p_{ij} defined by the formula below:

$$p_{ij} = Attr_{ij} \Big/ \sum_{d\in N_i} Attr_{id} \qquad if\ j \in N_i \tag{6.30}$$

And the attraction value of arc (i, j) for an ant is

$$Attr_{ij} = [\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta} \qquad (6.31)$$

Since the objective of this study is to mitigate the delay risk resulting from aircraft maintenance, it is more preferred to allocate more buffer time to each maintenance check. Therefore, if there is a maintenance check in between the arc (i, j), η_{ij} equals to $(DT_j - AT_i - MT_n)/(DT_j - AT_i)$. Otherwise, η_{ij} is calculated as $(DT_j - AT_i)/(DT_j - AT_i - TAT_k)$ to give the priority to the flight connection with less buffer time in order to increase aircraft utilization.

(2) The global pheromone trail update

After each ant completes its routes construction, the intensity of pheromone trails on each arc is updated using equation (31).

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \Delta\tau_{ij}^{best} \tag{6.32}$$

$$\Delta\tau_{ij}^{best} = Q/obj(S_{best}) \tag{6.33}$$

where the first term is used to decrease the pheromone values on all arcs by pheromone evaporation and the second term is used to increase the pheromone values on the arcs of the current best-found solution. The pheromone increments are calculated by the equation (6.32). $obj(S_{best})$ refers to the current best-found objective value. The parameter Q is used to adjust pheromone increments so that the search exploration and the search exploitation can be balanced.

6.5.2 A SP procedure

In ACO, a solution can be rejected due to its worse objective value in comparison with the current best solution. However, a rejected solution could also contain good routes, which can be used to form a better solution by matching the routes of different solutions. In this regard, we populate a pool of routes using the solutions derived from the ACO algorithm. By solving a SP model with the columns corresponding to the routes in the pool, an improved solution was produced. The SP model is described as below.

Let Ω be the pool of routes. We define z_r as binary variable that takes value of 1 if a route $r \in \Omega$ is selected, d_r as its total maintenance delay risks and α_{ir} as a binary parameter that takes value of 1 if fight leg $i \in F$ is covered by route $r \in \Omega$. Consider the following SP formulation:

$$min \sum_{r \in \Omega} z_r \, c_r \tag{6.34}$$

$$\sum_{r \in \Omega} z_r \alpha_{ir} = 1 \quad i \in F \tag{6.35}$$

$$z_r \in \{0,1\} \qquad r \in \Omega \tag{6.36}$$

The objective function (6.34) minimizes the summation of delay risks by selecting the best combination of routes. Constraint (6.35) states that each flight leg $i \in F$ must be covered once and only once.

6.5.3 A NS procedure

This phase is to explore the neighborhoods of the incumbent best-solution by iteratively solving the reduced instances using a commercial solver, i.e., CPLEX. To specify this phase in detail, the following notations are used. BS is defined as the incumbent best solution that consists of h aircraft routes $r_1, \ldots\ldots, r_h$. The reduced instance represented by $RS(RS \subseteq BS)$ is built by extracting a specific number of aircraft routes from the BS. This specific number is denoted by nr. nr_{min} and nr_{max} is the minimum and maximum possible value of nr, respectively. The NS approach is described as below:

Step 1: Starting from the$\boldsymbol{BS}$. Set the iteration number ϑ to be 1 and $nr = nr_{min}$.

Step 2: Constructing the $\boldsymbol{RS}$. The strategy for constructing the RS is specified as below:

d) Calculate the average delay risk score per route in BS, which is denoted by rl_{ave}, by dividing the total delay risk score by the number of routes.

e) Based on the value of rl_{ave}, the set of routes in BS is divided into two subsets κ and $\lambda\kappa$. The κ contains the routes in which the risk score is larger than the value of rl_{ave} while the $\lambda\kappa$ is its complementary. The number of routes in κ is μ_1 and in $\lambda\kappa$ is μ_2.

f) if $nr - 1 <= \mu_2$, randomly select $nr - 1$ routes in $\lambda\kappa$ and one route in κ to generate the RS, otherwise randomly select μ_2 routes in $\lambda\kappa$ and $nr - \mu_2$ routes in κ to generate the RS.

Step 3: Optimizing the $\boldsymbol{RS}$. Formulate the RS as a MILP model given in (1)-(27) and optimally solve the model.

Step 4: Updating the $\boldsymbol{BS}$. If a better solution is derived, then i) update the BS by substituting the old routes with the obtained solution, ii) set $\vartheta = \vartheta + 1$, iii) go to step

5, otherwise go to step 5**.**

Step 5: if $\vartheta <= \vartheta_{max}$, go to step 2, otherwise go to step 6.

Step 6: if $nr < nr_{max}$, set $nr = nr + 1$, $\vartheta = 1$ and proceed to step 2, otherwise output the BS and end.

6.5.4 Overview of the matheuristic approach

Based on the aforementioned components, a hybrid solution approach is designed, which combines the ACO algorithm, the SP process, and the NS procedure into a sequential scheme. We make two observations to enhance the efficiency of the proposed matheuristic. To begin, it is known that Mix Integer Programming (MIP) solver generally adopts a branch-and-bound or a branch-and-cut procedure. To increase the efficiency of the MIP solver, we improve its cutoff ability by utilizing the current best solution as the initial upper bound. Second, when the number of solutions in the pool hits a certain threshold, we update the pool by substituting the poorest solution for a better one. To avoid the inefficient updating, in ACO-SP-NS-b, if the SP procedure fails to produce a better solution based on the updated pool, this updating will be cleared. The outline of the proposed solution method is presented in algorithm 1.

Algorithm 1

```
Input: a set of flight legs $F$; a set of maintenance stations $M$; a set of aircraft $K$; number of iterations $NI$ and $\vartheta_{max}$; adjustable parameters $nr_{min}$, $nr_{max}$ ; maximum number of routes in routes pool $pn$
Output: $BS$
Initialization: $\Omega \leftarrow \Phi$; $BS \leftarrow \Phi$; $obj(BS) \leftarrow \infty$
for $j$ = 1 to $NI$ do
    Construct a complete solution (a set of aircraft maintenance routes) $S_j$ by using the state transition mechanism;
    if $obj(S_j)$< $obj(BS)$ then
        $BS \leftarrow S_j$;
    end if
    Update the pheromone trail by using the global pheromone trail update rule
    if $j <= pn$ do
        Add the routes of $S_j$ to $\Omega$;
    elseif $j > pn$ and $obj(S_j) < max(obj(S))$ (where $max(obj(S))$ represents the poorest solution in $\Omega$ which has the maximum objective value)
        Delete the routes of poorest solution $S$ in $\Omega$;
        Add the routes of $S_j$ to $\Omega$;
    end if
end for
Delete the duplicated routes in $\Omega$
$BS :=$ Solve the SP model over the set of routes stored in $\Omega$
for $nr = nr_{min}: nr_{max}$
    for $\vartheta = 1: \vartheta_{max}$
        Constructing the $RS$ using the routes of $BS$
        Optimizing the $RS$
        Updating the $BS$
    end for
end for
return $BS$
```

6.6 Computational experiments

In this section, computational experiments are carried out to test the performance of the proposed model. Our algorithms are implemented in MATLAB R2016a and the CPLEX 12.1is adopted to tackle the MILP model. All the experiments are carried out on a 2.60 GHz Intel® Core i7 processor and 16 GB RAM computer. In the following, Section 6.6.1 presents test instances used for the experiments. Then, section 6.6.2 compares the proposed model to the modified model that forbids violations of constraints on aircraft utilization. Finally, section 6.6.3 illustrates the impact of the risk-averse attitude of airlines on the performance of the proposed model.

6.6.1 Test instances

In the experiments, we collect twelve test instances with a one-month planning horizon from the BTS (BTS, 2020) database. They comprise twelve flight schedules with the number of flight legs ranging from 40 to 642. Notably, the connecting time between two flights is a crucial factor that directly impacts the risk level of each maintenance check, while the connecting time between long-haul flights is relatively longer than that between short-haul flights. Accordingly, we consider six instances of schedules with short-haul flights (flight time is within eight hours) and six instances of schedules with long-haul flights (flight time is between eight hours and sixteen hours). The overview of the test instances is shown in Table 6.3.

Table 6.3: Test instances

Test instances		Flight legs	Fleet size	Airports	Maintenance Stations
Short-haul flights	1	160	3	30	11
	2	222	4	25	9
	3	309	7	18	8
	4	452	11	32	12
	5	539	17	24	10
	6	642	22	31	7
Long-haul flight	1	40	3	4	4
	2	85	5	12	4
	3	166	6	10	6
	4	199	7	8	7
	5	251	13	15	9
	6	379	17	13	6

6.6.2 Comparison between the proposed model and the modified model

Increasing buffer time for delay risk mitigation will inevitably lead to a reduced aircraft utilization, necessitating the setting of constraints on aircraft utilization. In the proposed model, the constraints on aircraft utilization including the constraint on the unutilized flying time and the constraint on the number of required aircraft are treated as soft constraints. In order to investigate if one can improve the solution quality by incorporating the constraints on aircraft utilization as soft constraints in the RAMRP, we modify the proposed model (P-M) and solve the modified model (M-M) using the proposed matheuristic approach. The modification is made by forbidding violations of the constraints on aircraft utilization. By adopting the same risk matrix shown in figure 6.10, the solution details of two models are shown in Table 6.4 for the test instances with short-haul flights and Table 6.5 for the test instances with long-haul flights. Note that the upper bound of the unutilized flying time (480 minutes) for long-haul flights is set much larger than that (60 minutes) for short-haul flights due to the significantly longer flight time of long-haul flights.

As discussed in section 6.2, the robustness measure, i.e., the total risk score, is given in the third column of Table 6.4. The strategy for robustness enhancement is to provide a higher buffer time for the task package with a higher risk level at the expense of additional unutilized flying time or extra aircraft. Therefore, the total buffer time, the total unutilized flying time and the total number of required aircraft are provided in the Table 6.4. For two models, the number of maintenance checks may differ due to the varying aircraft utilization involved. In this sense, the number of maintenance checks, as well as the average value of each maintenance check for the risk score, buffer time, and unutilized flying time are recorded for a better understanding of how the model works.

Short-haul flights From the third column of Table 6.4, it is reasonable to observe a lower total risk score for P-M as compared to M-M. This is because P-M can further reduce the total delay risks by assigning higher buffer time to maintenance checks through allowing the use of extra aircraft or additional unutilized flying time as compared to M-M. Accordingly, we can also observe that the average buffer time and the average unutilized flying time for each maintenance check, as well as the total buffer time and the total unutilized flying time increase for P-M in all instances, while the number of required aircraft increases for P-M in instance 4, 5, and 6. Besides, it should be pointed out that the number of maintenance checks decreases with an increase in the number of required aircraft for P-M in instance 4, 5, and 6. In such a circumstance, it is worthwhile to note that the average risk score for each maintenance check is still lower for P-M compared with M-M, which further demonstrates the efficiency of P-M.

Long-haul flights As shown in Table 6.5, the comparison result between two models in instance 6 is comparable to what is observed in instances 4, 5, and 6 of short-haul flights. However, the comparison results vary in the remaining instances. In instance 1 and 2, it is interesting to find that the solutions of two models are identical. This indicates that when considering the constraints on aircraft utilization, the best solution can be achieved without violating these constraints for P-M. On the other hand, in instances 4 and 5, we can observe that P-M can obtain a lower total risk score with an equal number of required aircraft and less unutilized flying time in comparison with M-M. In instance 3, to achieve a lower total risk score, the number of required aircraft increases for P-M, however the unutilized flying time for each maintenance check decreases as compared to M-M. The reason behind these observations is that additional route options can be produced due to the permissibility of violating the constrains on aircraft utilization, thus proving the efficiency of considering these constraints as soft constrains.

Discussion As discussed above, the advantages of P-M have been validated from two perspectives for both short-haul flights and long-haul flights. On one hand, with an equal number of used aircraft, on average, P-M reduces the risk score by 30% while increasing the unutilized flying time by 25.72% compared with P-M (as shown in instances 1, 2, and 3 of Table 6.4). Obviously, the total delay risks decrease significantly at a price of an acceptable increase in the unutilized flying time. In addition, what is more encouraging is that P-M may even realize the reduction on the total delay risks with less unutilized flying time under certain circumstances, as compared to M-M (as shown in instances 4 and 5 of Table 6.5). On the other hand, even though there may be more aircraft in use and consequently fewer maintenance checks, the average unutilized flying time for each maintenance check and the total unutilized flying time may decrease in P-M as compared to M-M (as shown in instance 3 of Table 6.5). Moreover, the average risk score of each maintenance check is consistently lower for P-M. Therefore, it can be concluded that P-M can achieve the most desirable solution that involves the minimum total delay risks while having the least violation of constraint on unutilized flying time and requiring the least extra aircraft.

	N	L	M	H	VH
VH	M	H	H	H	H
H	M	M	H	H	H
M	L	M	M	H	H
L	L	L	M	M	H
VL	VL	L	L	M	M
U	VL	VL	L	L	M
R	VL	VL	VL	L	L

Figure 6.10: Risk matrix (probability categories: R: remote, U: unlikely, VL: very low, L: low, M: medium, H: high, VH: very high; severity categories: N: negligible, L: low, M: moderate, H: high, VH: very high; risk level: VL: very low, L: low, M: medium, H: high).

Table 6.4: Solution details of two models (short-haul flights)

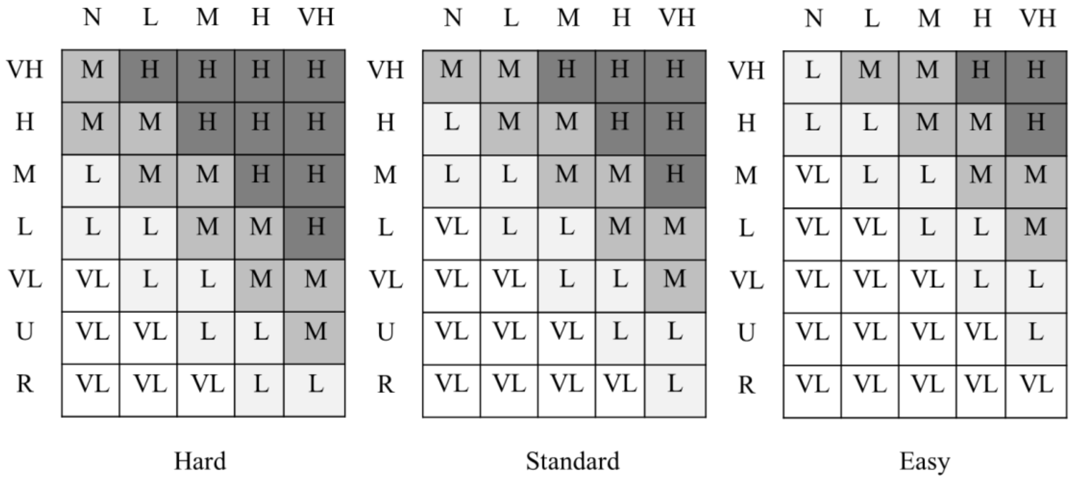
Instance	Model	Total risk score	Average risk score per check	Number of required aircraft	Number of checks	Total buffer time (minutes)	Average buffer time per check (minutes)	Total unutilized flying time (minutes)	Average unutilized flying time per check (minutes)
1	P-M	15.50	2.21	3	7	8,766	1,252	438	63
	M-M	23.54	3.36	3	7	6,260	894	375	54
2	P-M	23.01	2.56	4	9	22,959	2,551	641	71
	M-M	34.58	3.84	4	9	16,647	1,850	529	59
3	P-M	32.25	2.93	7	11	29,347	2,668	792	72
	M-M	42.15	3.83	7	11	22,995	2,090	569	52
4	P-M	49.26	3.08	13	16	29,318	1,832	845	53
	M-M	67.21	3.73	11	18	23,111	1,284	834	46
5	P-M	53.14	3.13	18	17	29,318	1,725	1,047	62
	M-M	64.27	3.57	17	18	10,616	590	951	53
6	P-M	56.73	2.70	24	21	32,406	1,543	1,218	58
	M-M	75.04	3.41	22	22	23,410	1,064	1,091	50

Table 6.5: Solution details of two models (long-haul flights)

Instance	Model	Total risk score	Average risk score per check	Number of required aircraft	Number of checks	Total buffer time (minutes)	Average buffer time per check (minutes)	Total unutilized flying time (minutes)	Average unutilized flying time per check (minutes)
1	P-M	17.43	2.49	3	7	860	123	2,822	403
	M-M	17.43	2.49	3	7	860	123	2,822	403
2	P-M	39.03	2.30	5	17	88,442	5,202	6,548	385
	M-M	39.03	2.30	5	17	88,442	5,202	6,548	385
3	P-M	76.51	2.19	7	35	320,718	9,163	13,575	388
	M-M	87.14	2.42	6	36	263,167	7,310	13,987	389
4	P-M	81.48	1.89	7	43	524,158	12,190	17,315	403
	M-M	96.77	2.25	7	43	436,811	10,158	17,318	403
5	P-M	115.60	2.18	13	53	640,223	12,080	21,154	399
	M-M	127.87	2.41	13	53	524,951	9,905	21,255	401
6	P-M	167.24	2.06	18	81	728,858	8,998	31,964	395
	M-M	188.67	2.30	17	82	642,179	7,831	27,799	339

6.6.3 The impact of the risk-averse attitude on the performance of the proposed model

An important factor that directly affects the decision-making on maintenance routes construction is the attitude of airlines regarding the delay risk resulting from aircraft maintenance. In the proposed model, we model the risk-averse attitude using the risk assessment matrix. For the same task package, applying the matrix with a greater category of risk could result in a higher risk level, suggesting a more risk-averse attitude. In order to evaluate how the risk-averse attitude affects the performance of the proposed model, we analyze three risk assessment matrices, namely easy matrix, standard matrix, and hard matrix, which provide lower, medium, and higher degrees of risk-aversion. The three risk assessment matrices are shown in Figure 6.11. The solution details utilizing the different risk assessment matrices are summarized in Table 6.6 for the test instances with short-haul flights and Table 6.7 for the test instances with long-haul flights. The notations in the Tables 6.6 and 6.7 are identical to the Tables 6.4 and 6.5.



	N	L	M	H	VH
VH	M	H	H	H	H
H	M	M	H	H	H
M	L	M	M	H	H
L	L	L	M	M	H
VL	VL	L	L	M	M
U	VL	VL	L	L	M
R	VL	VL	VL	L	L

Hard

	N	L	M	H	VH
VH	M	M	H	H	H
H	L	M	M	H	H
M	L	L	M	M	H
L	VL	L	L	M	M
VL	VL	VL	L	L	M
U	VL	VL	VL	L	L
R	VL	VL	VL	VL	L

Standard

	N	L	M	H	VH
VH	L	M	M	H	H
H	L	L	M	M	H
M	VL	L	L	M	M
L	VL	VL	L	L	M
VL	VL	VL	VL	L	L
U	VL	VL	VL	VL	L
R	VL	VL	VL	VL	VL

Easy

Figure 6.11: Different risk assessment matrices

Short-haul flights From Table 6.6, we can see that the total risk score and the average risk score increase with a higher degree of risk-aversion in all instances. This is because a higher risk level will be derived for the same combination of risk severity and risk probability by adopting a matrix with a higher degree of risk-aversion.

Despite that, comparison results by applying different risk matrices vary among the instances. In the instances 1, 2 and 3, adopting a matrix with a higher degree of risk-aversion increases the average buffer time and the average unutilized flying time for each maintenance check, as well as the total buffer time and the total unutilized flying time, whereas the number of required aircraft and the number of maintenance checks remain the same. With an increase in instance size, the number of required aircraft becomes larger when a matrix with a higher degree of risk-aversion is applied, as shown in instances 4, 5, and 6. In this case, the total buffer time and the total unutilized flying time may decrease due to the fewer maintenance checks; nonetheless, the rising trend for their average value of each maintenance check remains unchanged. The reason behind the above observations is intuitive. A higher risk-averse attitude indicates a greater willingness to pay for a further risk reduction. As such, the model is more inclined to allocate higher buffer times for maintenance checks to alleviate the total delay risks, resulting in either higher unutilized flying time or extra aircraft.

Long-haul flights As illustrated in instances 1 and 2 of Table 6.7, the solutions obtained by applying different risk matrices are identical. The reason behind this is that the time interval between two long-haul flights is considerably longer, occasionally exceeding ten hours, providing sufficient buffer time for task packages with the consideration of delay risks. However, as the instance size grows, for each aircraft, there will be more maintenance checks involving different task packages, including those with a higher delay risk level, resulting in varying performance when applying different risk matrices. In the remaining instances, the comparative results between different risk matrices are remarkably comparable to the observations in short-haul flights except for the comparison results regarding the unutilized flying time. As can be seen in instance 6, the total unutilized flying time and the average unutilized flying time of each maintenance check decreases for a hard matrix as compared to a standard matrix. This is reasonable since using additional aircraft allows for constructing more new maintenance routes.

Discussion On average, adopting a standard risk matrix can increase the average buffer time by 9.73% with an increase in the average unutilized flying time of 13.86%

and the number of required aircraft of 1.14% compared to adopting an easy risk matrix, while using a hard risk matrix can increase the average buffer time by 28.59% and 15.24% with an increase in the average unutilized flying time of 21.65% and 6% as well as the number of required aircraft of 4.64% and 3.43% compared to adopting an easy risk matrix and a standard risk matrix, respectively. From above observations, we find that the average buffer time for each maintenance check will increase when applying a matrix with a higher degree of risk-aversion, indicating that more layers of protection are provided to mitigate delay risks. However, additional unutilized flying time and extra aircraft will be induced, implying that airline should pay a higher price for a further risk reduction. Therefore, it can be concluded that applying the matrix with a higher degree of risk-aversion can better protect the schedule from maintenance disruptions, however at a higher expense, such as lower aircraft utilization. Moreover, it is worthwhile to note that using a standard matrix is less cost-effective than using an easy matrix, as the average buffer time increase is quite small in comparison to the increase in average unutilized flying time. However, there is no risk matrix which is “best” for every airline since it is a matter of airlines’ risk preference. Such investigations therefor provide a reference for airlines to achieve the desirable risk mitigation at the acceptable expense by designing an appropriate risk matrix that represents their risk-averse attitude.

Table 6.6: Solution detail of applying three risk matrices (short-haul flights)

Test instances	Risk matrix	Total risk score	Average risk score per check	Number of required aircraft	Number of maintenance checks	Total buffer time (minutes)	Average buffer time per check (minutes)	Total unutilized flying time (minutes)	Average unutilized flying time per check (minutes)
1	Easy	10.76	1.54	3	7	4,387	627	249	36
	Standard	14.07	2.01	3	7	6,280	897	379	54
	Hard	15.50	2.21	3	7	8,766	1,252	438	63
2	Easy	15.00	1.67	4	9	9,701	1,078	354	39
	Standard	18.86	2.10	4	9	12,163	1,351	517	57
	Hard	23.01	2.56	4	9	22,959	2,551	641	71
3	Easy	25.28	2.30	7	11	23,744	2,159	648	59
	Standard	30.55	2.78	7	11	25,128	2,284	742	67
	Hard	32.25	2.93	7	11	29,347	2,668	792	72
4	Easy	38.04	2.24	11	17	30,417	1,789	692	41
	Standard	43.04	2.69	12	16	28,842	1,803	837	52
	Hard	49.26	3.08	13	16	29,318	1,832	845	53
5	Easy	41.80	2.32	17	18	29,348	1,630	914	51
	Standard	49.02	2.72	17	18	30,719	1,707	966	54
	Hard	53.14	3.13	18	17	29,318	1,725	1,047	62
6	Easy	43.48	1.98	22	22	32,368	1,471	1,161	53
	Standard	53.49	2.55	23	21	30,975	1,475	1,132	54
	Hard	56.73	2.70	24	21	32,406	1,543	1,218	58

Table 6.7: Solution detail of applying three risk matrices (long-haul flights)

Test instances	Risk matrix	Total risk score	Average risk score per check	Number of required aircraft	Number of maintenance checks	Total buffer time (minutes)	Average buffer time per check (minutes)	Total unutilized flying time (minutes)	Average unutilized flying time per check (minutes)
1	Easy	10.76	1.59	3	7	860	123	2,822	403
	Standard	14.07	1.85	3	7	860	123	2,822	403
	Hard	15.50	2.49	3	7	860	123	2,822	403
2	Easy	15.00	1.48	5	17	88,442	5,202	6,548	385
	Standard	18.86	1.79	5	17	88,442	5,202	6,548	385
	Hard	23.01	2.30	5	17	88,442	5,202	6,548	385
3	Easy	25.28	1.49	6	36	292,120	8,114	12,970	360
	Standard	30.55	1.69	6	36	320,893	8,914	13,977	388
	Hard	32.25	2.19	7	35	320,718	9,163	13,575	388
4	Easy	38.04	1.41	7	43	436,675	10,155	16,120	375
	Standard	43.04	1.67	7	43	465,671	10,830	16,898	393
	Hard	49.26	1.89	7	43	524,158	12,190	17,315	403
5	Easy	41.80	1.41	13	53	524,201	9,891	20,703	391
	Standard	49.02	1.84	13	53	583,512	11,010	20,916	395
	Hard	53.14	2.18	13	53	640,223	12,080	21,154	399
6	Easy	43.48	1.51	17	82	642,741	7,935	31,596	390
	Standard	53.49	1.76	17	82	700,390	8,647	32,736	404
	Hard	56.73	2.06	18	81	728,858	8,889	31,964	390

6.7 Summary

Aircraft maintenance routing heavily relies on aircraft maintenance to ensure an aircraft complies with civil aviation safety regulations. However, in order to make things simpler, the majority of studies in the AMRP field consider maintenance checks that involve the same maintenance tasks and durations. A real-world circumstance may be violated by such simplification, leading to a significant discrepancy between the check's actual duration and its initial plan, which severely disrupts real operations. In this study, we present a novel RAMRP model that considers the duration of the maintenance check as the cause of disruption with the aim of alleviating the impact of check duration variability on aircraft maintenance routing. In the proposed model, aircraft maintenance checks are modeled in a more realistic way by breaking down an entire A-check program into multiple task packages, each with different maintenance tasks and durations. Then, the delay risk resulting from the check duration variability is precisely evaluated for each package by adopting the fuzzy risk assessment model. Based on the results of assessment, a robustness enhancement strategy is developed that assigns extra buffer time to task packages with higher risk levels. Finally, in order to reflect the total delay risks of the maintenance routes, a robustness measurement, referred as the total risk score, is proposed.

Using data from the Bureau of Transportation Statistics (BTS), we investigate the proposed model from two aspects, namely the restriction on aircraft utilization and the risk-averse attitude, both of which have a significant impact on the solution quality. The former limits the maximum allowable reduction in delay risks while the latter determines the willingness of the proposed model to pay for the mitigation of delay risks. The efficiency of the proposed model, which treats the constrains on aircraft utilization as soft constraints, is first demonstrated by computational results. Then, by modeling the risk-averse attitude using the risk assessment matrix, computational findings show how varying levels of risk-aversion affect the proposed model. Accordingly, guidelines are provided to airlines on how to design a risk assessment matrix that accurately reflects their risk-averse attitude and make the robustness enhancement cost-effective.

Chapter 7- Conclusion and Future Work

7.1 Conclusion

This thesis focuses on investigating the AMRP from two perspectives: the operational side and the robust side. From the operational side, we propose a new AMRP model that incorporates the cruise speed control. Then, an IACO algorithm is developed to solve the AMRP-CSC model. To solve the AMRP-CSC model more efficiently, a matheuristic approach that incorporates the exact method with the meta-heuristic method is designed. Regarding the robust side, we develop a new RAMRP model that takes maintenance checks into account as the source of disruption and seeks to mitigate the impact of check duration variability on aircraft maintenance routing.

Chapter 2 provides a brief literature review associated with airline schedule planning and cruise speed control. Among them, the literature on airline schedule planning focuses on four problems, i.e., FSP, FAP, AMRP, and CSP. Based on this literature review, the research gaps are identified finally.

Chapter 3 presents a new AMRP model that incorporates the cruise speed control. In the proposed model, each flight leg is assigned a cruise time window where several leg copies with different cruise times are placed. Then, by selecting the appropriate leg copy the cruise time can be optimized. The objective function is to minimize the sum of aircraft usage costs, idle time costs and fuel burn related costs, so that the trade-off between the aircraft utilization and fuel burn related costs can be examined. In order to evaluate the impact of flexible cruise time on solution improvement, the proposed model is compared with the traditional model.

Chapter 4 proposes an IACO algorithm to effectively tackle AMRP-CSC. The traditional ACO algorithm selects the next covered flight leg depending on the attractiveness of flight connections, while ignoring the information regarding the individual flight leg, which fails to optimize the cruise times. Moreover, its performance dramatically decreases in tackling large-scale problems, particularly when modeling flexible cruise times causes an explosion in problem size. Therefore, we develop an improved ACO (IACO) algorithm with a new state transition mechanism incorporating the node-based heuristic information that provides the guidance for the optimization of cruise times, and a new pheromone updating mechanism to enhance the search efficiency and precision. Using the data from the Bureau of Transportation Statistics, we prove that the proposed algorithm reaches the optimality for small-size problems and outperforms the existing meta-heuristic approaches for large-size problems.

Chapter 5, based on the IACO algorithm presented in Chapter 4, introduces a novel solution methodology, i.e., a matheuristic approach, for AMRP-CSC. It is composed of three main components: an improved ant colony optimization (IACO) algorithm, a set partitioning (SP) procedure and a neighborhoods search (NS) procedure. The IACO algorithm serves as a route generator, populating a pool of routes with promising feasible aircraft maintenance routes. Then, a SP model, which features the high-quality columns corresponding to the routes in the pool, is solved to produce a possible better solution. Finally, this solution is further improved by a NS procedure that iteratively solves the reduced AMRP-CSC instances to optimality. This matheuristic approach is analyzed and test using the data extracting from the Bureau of Transportation Statistics (BTS), and then its accuracy and the efficiency have been demonstrated by experiment analyses.

Chapter 6 propose a new robust aircraft maintenance routing problem (RAMRP) model that takes maintenance checks into account as the source of disruption and seeks to alleviate the impact of check duration variability on aircraft maintenance routing. A matheuristic approach is developed for solving the proposed model. Using the data from the Bureau of Transportation Statistics (BTS), computational experiments are conducted to investigate the proposed model from two perspectives, namely the constraints on aircraft utilization and the risk-averse attitude of airlines, of which the former limits the maximum allowable reduction in delay risks and the latter determines the willingness of the proposed model to pay more for a further mitigation of delay risks. Thus, we demonstrate the advantage of the proposed model and highlights the significance of being properly risk-averse in decision making.

In conclusion, this thesis achieves two primary aims by incorporating the cruise speed control with the AMRP. The first aim is to improve aircraft utilization and reduce the number of aircraft required, while the second aim is to maintain schedule stability and flexibility. Despite the consideration of robustness in the AMRP-CSC, the robustness has not been quantified and the aircraft utilization improvement is the mainly focus. However, with an increasing emphasis on schedule robustness, this thesis focuses on developing a novel RAMRP model that can quantify the robustness. The proposed RAMRP model considers the duration of the maintenance check as the cause of disruption and thus achieves the robustness by alleviating the disruption risk arising from aircraft maintenance.

7.2 Future work

Despite the contributions made in our study, some limitations are identified. Based on these limitations, some future works are suggested as below:

1. Traditionally, the airline scheduling problem is decomposed into several stages and solved sequentially for the reasons of tractability. However, this approach fails to capture the dependencies between different stages. In the aircraft maintenance routing problem, one of the most crucial decisions imposed by the prior stages that impact on the aircraft route decision most is the scheduled block time determined in the flight scheduling stage. Therefore, the integration of flight schedule problem and AMRP problem will be another future direction.

2. In this research work, the cruise speed decisions are limited to the discrete

variables. However, considering the continuous options of cruise speed can obtain better solutions, which will simultaneously result in the considerably large problem size. To maximize aircraft utilization while minimizing fuel-burn related costs, the AMRP model will include the cruise speed as continuous variables and an effective method will be developed for it in the future.

3. To better capture the uncertainties in aircraft maintenance, in the future, we can incorporate the big data analytic techniques with the proposed RAMRP model. In this way, we can more comprehensively assess the risk resulting from aircraft maintenance in a dynamic and stochastic operating environment.

References

Abara, J. 1989. Applying integer linear programming to the fleet assignment problem. *Interfaces* 19 (4):20-28.

Ackert, S. P. 2010. Basics of aircraft maintenance programs for financiers. *Evaluation insights of commercial aircraft maintenance programs* 10.

Ahmed, M. B., F. Z. Mansour, and M. Haouari. 2017. A two-level optimization approach for robust aircraft routing and retiming. *Computers & Industrial Engineering* 112:586-594.

Aktürk, M. S., A. Atamtürk, and S. Gürel. 2014. Aircraft rescheduling with cruise speed control. *Operations Research* 62 (4):829-845.

Al-Thani, N. A., M. B. Ahmed, and M. Haouari. 2016. A model and optimization-based heuristic for the operational aircraft maintenance routing problem. *Transportation Research Part C: Emerging Technologies* 72:29-44.

Antunes, D., V. Vaze, and A. P. Antunes. 2019. A robust pairing model for airline crew scheduling. *Transportation Science* 53 (6):1751-1771.

Arıkan, U., S. Gürel, and M. S. Aktürk. 2016. Integrated aircraft and passenger recovery with cruise time controllability. *Annals of Operations Research* 236 (2):295-317.

Arıkan, U., S. Gürel, and M. S. Aktürk. 2017. Flight network-based approach for integrated airline recovery with cruise speed control. *Transportation Science* 51 (4):1259-1287.

Avci, M., and S. Topaloglu. 2016. A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications* 53:160-171.

Balseiro, S. R., I. Loiseau, and J. Ramonet. 2011. An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. *Computers & Operations Research* 38 (6):954-966.

Barnhart, C., N. L. Boland, L. W. Clarke, E. L. Johnson, G. L. Nemhauser, and R. G. Shenoi. 1998. Flight string models for aircraft fleeting and routing. *Transportation Science* 32 (3):208-220.

Barnhart, C., T. S. Kniker, and M. Lohatepanont. 2002. Itinerary-based airline fleet assignment. *Transportation Science* 36 (2):199-217.

Başdere, M., and Ü. Bilge. 2014. Operational aircraft maintenance routing problem with remaining time consideration. *European journal of operational research* 235 (1):315-328.

Belobaba, P. 2009. The airline planning process. *The global airline industry*:153-181.

Berge, M. E., and C. Hopperstad. 1993. Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Operations Research* 41 (1):153-168.

Bertazzi, L., L. C. Coelho, A. De Maio, and D. Laganà. 2019. A matheuristic algorithm for the multi-depot inventory routing problem. *Transportation Research Part E: Logistics and Transportation Review* 122:524-544.

Bourjade, S., R. Huc, and C. Muller-Vibes. 2017. Leasing and profitability: Empirical evidence from the airline industry. *Transportation Research Part A: Policy and Practice* 97:30-46.

BTS. 2020. Airline on-time performance data.

Bulbul, K. G., and R. Kasimbeyli. 2021. Augmented Lagrangian based hybrid subgradient method for solving aircraft maintenance routing problem. *Computers & Operations Research* 132:105294.

Burke, E. K., P. De Causmaecker, G. De Maere, J. Mulder, M. Paelinck, and G. V. Berghe. 2010. A multi-objective approach for robust airline scheduling. *Computers & Operations Research* 37 (5):822-832.

Cappanera, P., and G. Gallo. 2004. A multicommodity flow approach to the crew rostering problem. *Operations Research* 52 (4):583-596.

Chambari, A., P. Azimi, and A. A. Najafi. 2021. A bi-objective simulation-based optimization algorithm for redundancy allocation problem in series-parallel systems. *Expert Systems with Applications* 173:114745.

Chen, W.-T., K. Huang, and M. N. Ardiansyah. 2018. A mathematical programming model for aircraft leasing decisions. *Journal of Air Transport Management* 69:15-25.

Choi, T.-M., X. Wen, X. Sun, and S.-H. Chung. 2019. The mean-variance approach for global supply chain risk analysis with air logistics in the blockchain technology era. *Transportation Research Part E: Logistics Transportation Review* 127:178-191.

Clarke, L., E. Johnson, G. Nemhauser, and Z. Zhu. 1997. The aircraft rotation problem. *Annals of Operations Research* 69:33-46.

Cook, A. J., G. Tanner, and S. Anderson. 2004. Evaluating the true cost to airlines of one minute of airborne or ground delay.

Cui, R., X. Dong, and Y. Lin. 2019. Models for aircraft maintenance routing problem with consideration of remaining time and robustness. *Computers & Industrial Engineering* 137:106045.

Deng, G.-F., and W.-T. Lin. 2011. Ant colony optimization-based algorithm for airline crew scheduling problem. *Expert Systems with Applications* 38 (5):5787-5793.

Deng, Q., B. F. Santos, and R. Curran. 2020. A practical dynamic programming based methodology for aircraft maintenance check scheduling optimization. *European journal of operational research* 281 (2):256-273.

Desaulniers, G., J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M. M. Solomon, and F. Soumis. 1997a. Crew pairing at air france. *European journal of operational research* 97 (2):245-259.

Desaulniers, G., J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. 1997b. Daily aircraft routing and scheduling. *Management Science* 43 (6):841-855.

Deveci, M., and N. C. Demirel. 2018. Evolutionary algorithms for solving the airline crew pairing problem. *Computers & Industrial Engineering* 115:389-406.

Díaz-Ramírez, J., J. I. Huertas, and F. Trigos. 2014. Aircraft maintenance, routing, and crew scheduling planning for airlines with a single fleet and a single maintenance and crew base. *Computers & Industrial Engineering* 75:68-78.

Doerner, K. F., and V. Schmid. 2010. Survey: matheuristics for rich vehicle routing problems. Paper read at International Workshop on Hybrid Metaheuristics.

Dong, Z., Y. Chuhang, and H. H. Lau. 2016. An integrated flight scheduling and fleet assignment method based on a discrete choice model. *Computers & Industrial Engineering* 98:195-210.

Dumas, J., F. Aithnard, and F. Soumis. 2009. Improving the objective function of the fleet assignment problem. *Transportation Research Part B: Methodological* 43 (4):466-475.

Dunbar, M., G. Froyland, and C.-L. Wu. 2012. Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. 46 (2):204-216.

Dunbar, M., G. Froyland, C.-L. J. C. Wu, and O. Research. 2014. An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. 45:68-86.

Duran, A. S., S. Gürel, and M. S. Aktürk. 2015. Robust airline scheduling with controllable cruise times and chance constraints. *IIE Transactions* 47 (1):64-83.

Eltoukhy, A. E., F. T. Chan, S. Chung, and T. Qu. 2017a. Scenario-based Stochastic Framework for Operational Aircraft Maintenance Routing Problem. Paper read at The International Conference on Systems Engineering and Engineering Management (ICSEEM'17).

Eltoukhy, A. E., F. T. Chan, S. H. Chung, and B. Niu. 2018a. A model with a solution algorithm for the operational aircraft maintenance routing problem. *Computers & Industrial Engineering* 120:346-359.

Eltoukhy, A. E., F. T. Chan, S. H. Chung, B. Niu, and X. Wang. 2017b. Heuristic approaches for operational aircraft maintenance routing problem with maximum flying hours and man-power availability considerations. *Industrial Management Data Systems*.

Eltoukhy, A. E., Z. Wang, F. T. Chan, S. Chung, H.-L. Ma, and X. Wang. 2019. Robust Aircraft Maintenance Routing Problem Using a Turn-Around Time Reduction Approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems.*

Eltoukhy, A. E., Z. Wang, F. T. Chan, and S. H. Chung. 2018b. Joint optimization using a leader–follower Stackelberg game for coordinated configuration of stochastic operational aircraft maintenance routing and maintenance staffing. *Computers & Industrial Engineering* 125:46-68.

EUROCONTROL. 2001. Forecasting civil aviation fuel burn and emissions in Europe. *Technical Report 2001-8. EEC Technical/Scientific, Eurocontrol, Eurocontrol Ex- perimental Centre, B.P. 15, F-91222 Bretigny-sur-Orge, France .*

EUROCONTROL. 2013. User manual for the base of aircraft data (BADA) revision 3.11. *Technical Report 13/04/16-01. EEC Technical/Scientific, Eurocontrol, Euro- control Experimental Centre, B.P. 15, F-91222 Bretigny-sur-Orge, France.*

Gao, C., E. Johnson, and B. Smith. 2009. Integrated airline fleet and crew robust planning. *Transportation Science* 43 (1):2-16.

Gopalan, R., and K. Talluri. 1998. The aircraft maintenance routing problem. *Operations Research* 46 (2):260-271.

Gürkan, H., S. Gürel, and M. S. Aktürk. 2016. An integrated approach for airline scheduling, aircraft fleeting and routing with cruise speed control. *Transportation Research Part C: Emerging Technologies* 68:38-57.

Hane, C. A., C. Barnhart, E. L. Johnson, R. E. Marsten, G. L. Nemhauser, and G. Sigismondi. 1995. The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming* 70 (1-3):211-232.

Haouari, M., S. Shao, and H. Sherali. 2013. A lifted compact formulation for the daily aircraft maintenance routing problem. *Transportation Science* 47 (4):508-525.

Haouari, M., H. D. Sherali, F. Z. Mansour, and N. Aissaoui. 2011. Exact approaches for integrated aircraft fleeting and routing at TunisAir. *Computational Optimization and Applications* 49 (2):213-239.

Hooplot, J. N., and A. Ghobbar. 2010. Redesigning Maintenance Processes to Increase Delivery Performance of the A-check for Wide-body Aircraft at KLM E&M. In *Air Transport and Operations*: IOS Press, 1-11.

Huang, S.-H., Y.-H. Huang, C. A. Blazquez, and G. Paredes-Belmar. 2018. Application of the ant colony optimization in the resolution of the bridge inspection routing problem. *Applied soft computing* 65:443-461.

Jacobs, T. L., B. C. Smith, and E. L. Johnson. 2008. Incorporating network flow effects into the airline fleet assignment process. *Transportation Science* 42 (4):514-529.

Jalalian, M., S. Gholami, and R. Ramezanian. 2019. Analyzing the trade-off between CO2 emissions and passenger service level in the airline industry: Mathematical modeling and constructive heuristic. *Journal of Cleaner Production.* 206:251-266.

Jamili, A. 2017. A robust mathematical model and heuristic algorithms for integrated aircraft routing and scheduling, with consideration of fleet assignment problem. *Journal of Air Transport Management* 58:21-30.

Jiang, H., and C. Barnhart. 2009. Dynamic airline scheduling. *Transportation Science* 43 (3):336-354.

Jiang, H., and C. Barnhart. 2013. Robust airline schedule design in a dynamic scheduling environment. *Computers & Operations Research* 40 (3):831-840.

Kabbani, N. M., and B. W. Patty. 1992. Aircraft routing at American airlines. Paper read at proceedings of the agifors symposium.

Kenan, N., A. Jebali, and A. Diabat. 2018. An integrated flight scheduling and fleet assignment problem under uncertainty. *Computers & Operations Research* 100:333-342.

Klabjan, D., E. L. Johnson, G. L. Nemhauser, E. Gelman, and S. Ramaswamy. 2001. Airline crew scheduling with regularity. *Transportation Science* 35 (4):359-374.

Kohl, N., and S. E. Karisch. 2004. Airline crew rostering: Problem types, modeling, and optimization. *Annals of Operations Research* 127 (1-4):223-257.

Kramer, R., A. Subramanian, T. Vidal, and L. d. A. F. Cabral. 2015. A matheuristic approach for the Pollution-Routing Problem. *European journal of operational research* 243 (2):523-539.

Lan, S., J.-P. Clarke, and C. Barnhart. 2006. Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science* 40 (1):15-28.

Lee, L. H., C. U. Lee, and Y. P. Tan. 2007. A multi-objective genetic algorithm for robust flight scheduling using simulation. *European Journal of Operational Research* 177 (3):1948-1968.

Leggieri, V., and M. Haouari. 2018. A matheuristic for the asymmetric capacitated vehicle routing problem. *Discrete Applied Mathematics* 234:139-150.

Liang, Z., and W. A. Chaovalitwongse. 2013. A network-based model for the integrated weekly aircraft maintenance routing and fleet assignment problem. *Transportation Science* 47 (4):493-507.

Liang, Z., W. A. Chaovalitwongse, H. C. Huang, and E. L. Johnson. 2011. On a new rotation tour network model for aircraft maintenance routing problem. *Transportation Science* 45 (1):109-120.

Liang, Z., Y. Feng, X. Zhang, T. Wu, and W. A. Chaovalitwongse. 2015. Robust weekly aircraft maintenance routing problem and the extension to the tail assignment problem. *Transportation Research Part B: Methodological* 78:238-259.

Lin, S.-W., C.-Y. Cheng, P. Pourhejazy, K.-C. Ying, and C.-H. Lee. 2021. New benchmark algorithm for hybrid flowshop scheduling with identical machines. *Expert Systems with Applications* 183:115422.

Lohatepanont, M., and C. Barnhart. 2004. Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science* 38 (1):19-32.

Machado, A. M., G. R. Mauri, M. C. S. Boeres, and R. d. A. Rosa. 2021. A new hybrid matheuristic of GRASP and VNS based on constructive heuristics, set-covering and set-partitioning formulations applied to the capacitated vehicle routing problem. *Expert Systems with Applications* 184:115556.

Maenhout, B., and M. Vanhoucke. 2010. A hybrid scatter search heuristic for personalized crew rostering in the airline industry. *European journal of operational research* 206 (1):155-167.

Maher, S. J., G. Desaulniers, and F. Soumis. 2014. Recoverable robust single day aircraft maintenance routing problem. *Computers & Operations Research* 51:130-145.

Mancini, S. 2016. A real-life Multi Depot Multi Period Vehicle Routing Problem with a Heterogeneous Fleet: Formulation and Adaptive Large Neighborhood Search based Matheuristic. *Transportation Research Part C: Emerging Technologies* 70:100-112.

Marais, K., and I. A. Waitz. 2009. Air transport and the environment. *The global airline industry*:405-440.

Markowski, A. S., and M. S. Mannan. 2008. Fuzzy risk matrix. *Journal of hazardous materials* 159 (1):152-157.

Marla, L., B. Vaaben, and C. Barnhart. 2017. Integrated disruption management and flight planning to trade off delays and fuel burn. *Transportation Science* 51 (1):88-111.

Mercier, A., J.-F. Cordeau, and F. Soumis. 2005. A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research* 32 (6):1451-1476.

Moussavi, S. E., M. Mahdjoub, and O. Grunder. 2019. A matheuristic approach to the integration of worker assignment and vehicle routing problems: Application to home healthcare scheduling. *Expert Systems with Applications* 125:317-332.

Muter, İ., Ş. İ. Birbil, K. Bülbül, G. Şahin, H. Yenigün, D. Taş, and D. Tüzün. 2013. Solving a robust airline crew pairing problem with column generation. *Computers & Operations Research* 40 (3):815-830.

Oum, T. H., A. Zhang, and Y. Zhang. 2000. Optimal demand for operating lease of aircraft. *Transportation Research Part B: Methodological* 34 (1):17-29.

Phoemphon, S., C. So-In, and N. Leelathakul. 2021. Improved distance estimation with node selection localization and particle swarm optimization for obstacle-aware wireless sensor networks. *Expert Systems with Applications* 175:114773.

Pilla, V. L., J. M. Rosenberger, V. Chen, N. Engsuwan, and S. Siddappa. 2012. A multivariate adaptive regression splines cutting plane approach for solving a two-stage stochastic programming fleet assignment model. *European journal of operational research* 216 (1):162-171.

Pilla, V. L., J. M. Rosenberger, V. C. Chen, and B. Smith. 2008. A statistical computer experiments approach to airline fleet assignment. *IIE Transactions* 40 (5):524-537.

Pita, J. P., C. Barnhart, and A. P. Antunes. 2013. Integrated flight scheduling and fleet assignment under airport congestion. *Transportation Science* 47 (4):477-492.

Queiroga, E., R. Sadykov, and E. Uchoa. 2021. A POPMUSIC matheuristic for the capacitated vehicle routing problem. *Computers & Operations Research* 136:105475.

Quesnel, F., G. Desaulniers, and F. Soumis. 2020. Improving air crew rostering by considering crew preferences in the crew pairing problem. *Transportation Science* 54 (1):97-114.

Rexing, B., C. Barnhart, T. Kniker, A. Jarrah, and N. Krishnamurthy. 2000. Airline fleet assignment with time windows. *Transportation Science* 34 (1):1-20.

Rosales, L. J. S., J.-B. Yang, and Y.-W. Chen. 2014. Analysing delays and disruptions in Aircraft Heavy Maintenance. Paper read at The 32nd International Conference of the System Dynamics Society. Delft, Netherlands.

Rosenberger, J. M., E. L. Johnson, and G. L. Nemhauser. 2004. A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science* 38 (3):357-368.

Ruan, J. H., Z. X. Wang, F. T. S. Chan, S. Patnaik, and M. K. Tiwari. 2021. A reinforcement learning-based algorithm for the aircraft maintenance routing problem. *Expert Systems with Applications* 169:114399.

Rushmeier, R. A., and S. Kontogiorgis. 1997. Advances in the optimization of airline fleet assignment. *Transportation Science* 31 (2):159-169.

Saddoune, M., G. Desaulniers, I. Elhallaoui, and F. Soumis. 2012. Integrated airline crew pairing and crew assignment by dynamic constraint aggregation. *Transportation Science* 46 (1):39-55.

Şafak, Ö., S. Gürel, and M. S. Aktürk. 2017. Integrated aircraft-path assignment and robust schedule design with cruise speed control. *Computers & Operations Research* 84:127-145.

Salazar-González, J.-J. 2014. Approaches to solve the fleet-assignment, aircraft-routing, crew-pairing and crew-rostering problems of a regional carrier. *Omega* 43:71-82.

Sandhu, R., and D. Klabjan. 2007. Integrated airline fleeting and crew-pairing decisions. *Operations Research* 55 (3):439-456.

Sarac, A., R. Batta, and C. M. Rump. 2006. A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research* 175 (3):1850-1869.

Shao, S., H. D. Sherali, and M. Haouari. 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.

Shebalov, S., and D. Klabjan. 2006. Robust airline crew pairing: Move-up crews. *Transportation Science* 40 (3):300-312.

Sherali, H. D., K.-H. Bae, and M. Haouari. 2013. A benders decomposition approach for an integrated airline schedule design and fleet assignment problem with flight retiming, schedule balance, and demand recapture. *Annals of Operations Research* 210 (1):213-244.

Skorupski, J. 2016. The simulation-fuzzy method of assessing the risk of air traffic accidents using the fuzzy risk matrix. *Safety science* 88:76-87.

Smith, B. C., and E. L. Johnson. 2006. Robust airline fleet assignment: Imposing station purity using station decomposition. *Transportation Science* 40 (4):497-516.

Sohoni, M., Y.-C. Lee, and D. Klabjan. 2011. Robust airline scheduling under block-time uncertainty. *Transportation Science* 45 (4):451-464.

Sriram, C., and A. Haghani. 2003. An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A: Policy and Practice* 37 (1):29-48.

Sun, X., S. H. Chung, and H. L. Ma. 2020. Operational risk in airline crew scheduling: do features of flight delays matter? *Decision Sciences*.

Talluri, K. T. 1998. The four-day aircraft maintenance routing problem. *Transportation Science* 32 (1):43-53.

van der Weide, T., Q. Deng, and B. F. Santos. 2022. Robust long-term aircraft heavy maintenance check scheduling optimization under uncertainty. *Computers & Operations Research* 141:105667.

Wang, K., Y. Shao, and W. Zhou. 2017. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. *Transportation Research Part D-transport Environment* 57:262-276.

Weide, O., D. Ryan, and M. Ehrgott. 2010. An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research* 37 (5):833-844.

Yan, C., and J. Kung. 2018. Robust aircraft routing. *Transportation Science* 52 (1):118-133.

Yan, S., and J.-C. Chang. 2002. Airline cockpit crew scheduling. *European journal of operational research* 136 (3):501-511.

Yan, S., C.-H. Tang, and T.-C. Fu. 2008. An airline scheduling model and solution algorithms under stochastic demands. *European journal of operational research* 190 (1):22-39.

Yan, S., C.-H. Tang, and M.-C. Lee. 2007. A flight scheduling model for Taiwan airlines under market competitions. *Omega* 35 (1):61-74.

Yan, S., and C.-H. Tseng. 2002. A passenger demand model for airline flight scheduling and fleet routing. *Computers & Operations Research* 29 (11):1559-1581.

Yan, S., and H.-F. Young. 1996. A decision support framework for multi-fleet routing and multi-stop flight scheduling. *Transportation Research Part A: Policy and Practice* 30 (5):379-398.

Zámková, M., M. Prokop, and R. Stolín. 2017. Factors influencing flight delays of a European airline. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis* 65 (5):1799-1807.

Zeighami, V., and F. Soumis. 2019. Combining Benders’ decomposition and column generation for integrated crew pairing and personalized crew assignment problems. *Transportation Science* 53 (5):1479-1499.

Zeren, B., and I. Özkol. 2016. A novel column generation strategy for large scale airline crew pairing problems. *Expert Systems with Applications* 55:133-144.

Zhang, K., M. Duan, X. Luo, and G. Hou. 2017. A fuzzy risk matrix method and its application to the installation operation of subsea collet connector. *Journal of Loss Prevention in the Process Industries* 45:147-159.

Zhou, S.-Z., Z.-H. Zhan, Z.-G. Chen, S. Kwong, and J. Zhang. 2020. A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction. *IEEE Transactions on Intelligent Transportation*

Systems.