OPTIMIZING AIRCRAFT ROUTING OF AIRLINE AND MAINTENANCE STAFFING OF MAINTENANCE PROVIDERS USING GAME THEORETIC MODEL

ELSAYED ELSAYED ELTOUKHY ABDELRAHMAN

PhD

The Hong Kong Polytechnic University

2019

The Hong Kong Polytechnic University

Department of Industrial and Systems Engineering

Optimizing Aircraft Routing of Airline and Maintenance Staffing of Maintenance providers Using Game Theoretic Model

ELSAYED ELSAYED ELTOUKHY Abdelrahman

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

July 2018

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)

Abdelrahman ELSAYED ELSAYED ELTOUKHY (Name of student)

Abstract

Recently, aviation and airline maintenance providers are of the most significant worldwide industries. This is shown by the enormous growth in the number of passengers, which was around 3.5 billion passengers in 2015, expecting an annual growth of 5%. To cope with this passenger growth, the number of aircraft is expected to increase from 24,579 in 2014 to 29,955 in 2022. As a result, the aircraft maintenance cost paid by airlines to maintenance providers is expected to increase from US $62.1 billion in 2014 to US $90 billion in 2024. Despite this pleasing economic situation for airlines and maintenance providers, many difficult challenges have been emerged during the planning and operating processes. One of the challenges facing airlines is how to build efficient routes for their aircraft, while respecting the operational maintenance restrictions. In this regard, aircraft maintenance routing problem (AMRP) is very significant for airlines, as it builds the routes for their aircraft and schedule their maintenance visits. On the other hand, for the maintenance providers, it is a great challenge to manage the workforce capacity required to serve the increased number of aircraft. Therefore, maintenance staffing problem (MSP) is recognized as an effective tool for maintenance providers, as it manages the workforce capacity required to serve the airlines' aircraft.

In the existing research, on the focus of AMRP, most AMRP models consider one operational maintenance restriction, which is a single maintenance visit every four days and overlook the restrictions of the total cumulative flying time, the total number of take-offs, the workforce capacity and the working hours of the maintenance providers. Consequently, the generated routes are not applicable in real practice due to their

infeasibility. This motivates us to develop a model, in which all the aforementioned restrictions are considered in a single model. Therefore, the routes determined by this model can be implemented in reality. In addition, an efficient solution algorithm is proposed for solving the developed model. Meanwhile, one of the glaring facts in the literature is that AMRP and MSP are studied independently, and their interdependence have not been investigated. To fulfil this research gap, a leader-follower Stackelberg game (LFSG) model is developed to capture this interdependence. Moreover, a nested ant colony optimization-based algorithm is proposed as a solution method for the game theoretic model. Towards the goal of showing the superiority of the proposed model, we present a case study of LFSG for one major airline and four maintenance providers located in the Middle East. The results show significant cost savings for all players.

Although the LFSG presents a formulation for a unique problem in the literature, it overlooks one important aspect, called the price competition among the maintenance providers. Indeed, this aspect has a direct influence on the AMRP, as it changes the routing plan constructed by airlines. In this connection, it is imperative to consider the price competition among the maintenance providers besides the interdependence between AMRP and MSP. For this purpose, a Stackelberg-Nash game model (SNGM) is proposed to capture the above-mentioned problem. In addition, an iterative game algorithm is developed in order to obtain the overall Nash equilibrium for the SNGM. To demonstrate the viability of the proposed model, we use the previous case study, in which its results reveal significant savings for airline and maintenance providers.

The contribution of this thesis is threefold. Firstly, proposing a new scalable AMRP that considers all the operational maintenance restrictions along with developing an efficient solution algorithm to solve this model. Secondly, developing a coordinated decision support system based on game theory to capture the interdependence between AMRP of airlines and MSP of maintenance providers. Lastly, modeling the previous interdependence in the presence of the price competition among the maintenance providers, we develop a new model, called Stackelberg-Nash game model.

Publications

International Journal

Eltoukhy, A.E.E., Chan, F.T.S. and Chung, S.H., 2017. Airline schedule planning: A review and future directions. *Industrial Management & Data Systems*, 117(6), 1201-1243.

Eltoukhy, A.E.E., Chan, F.T.S., Chung, S.H., Niu, B. and Wang, X.P., 2017. Heuristic approaches for operational aircraft maintenance routing problem with maximum flying hours and man-power availability considerations. *Industrial Management & Data Systems*, 117(10), 2142-2170.

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

Eltoukhy, A.E.E., Wang, Z.X., Chan, F.T.S. and Chung, S.H., 2018. 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.

Eltoukhy, A.E.E., Wang, Z.X., Chan, F.T.S. and Ruan, J.H., 2018. Data analytics in optimizing aircraft routing and maintenance staffing with price competition consideration by a Stackelberg-Nash game model. *Transportation Research Part E: Logistics and Transportation Review*. (Under second review)

International Conference

Eltoukhy, A.E.E., Chan, F.T.S. and Chung, S.H., 2016. Ant colony optimization and simulated annealing for aircraft maintenance routing problem. *The 26th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM)*, Seoul, Korea, 27-30 June 2016.

Eltoukhy, A.E.E., Chan, F.T.S., Chung, S.H. and Qu, T., 2017. Optimization Model and Solution Method for Operational Aircraft Maintenance Routing Problem. *World Congress on Engineering 2017*, London, UK, 5-7 July 2017.

Eltoukhy, A.E.E., Chan, F.T.S., Chung, S.H. and Qu, T., 2017. Stackelberg game theoretical model for optimizing aircraft maintenance routing with maintenance staffing. *The 47th International Conference on Computers & Industrial Engineering*, Lisbon, Portugal, 11-13 October 2017.

Eltoukhy, A.E.E., Chan, F.T.S., Chung, S.H. and Qu, T., 2017. Scenario-based Stochastic Framework for Operational Aircraft Maintenance Routing Problem. *International Conference on Systems Engineering and Engineering Management (ICSEEM'17)*, San Francisco, USA, 25-27 October 2017.

Chan, F.T.S., **Eltoukhy, A.E.E.**, 2018. Investigating the interrelationship between stochastic aircraft routing of airlines and maintenance staffing of maintenance providers. *The 5th International Conference on Industrial Engineering and Applications (ICIEA 2018)*, Singapore, 26-28 April 2018.

Eltoukhy, A.E.E., Chan, F.T.S., Chung, S.H. and Niu, B., 2018. Compressed annealing-based algorithm for operational aircraft maintenance routing problem with the consideration of working hours of maintenance stations. *Global Conference on Engineering and Applied Science (GCEAS 2018)*, Tokyo, Japan, 10-12 July 2018.

Acknowledgement

Firstly, I would like to express my sincere gratitude to my supervisor, Professor Felix T.S. Chan, for his advice, patience, support, guidance and encouragement during my PhD period. I further wish to thank Dr. S.H. Chung for his immense knowledge provided to my research work. Honestly, without his help and advice, I would not be able to achieve this far.

Secondly, I would like to thank the support from ISE department, especially Prof. Yue, Ms. Cammy Chiu, and Ms. Iris Ko.

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

Thank you for your support. Only my god could repay all your help and kindness.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AMRP	Aircraft Maintenance Routing Problem
MSP	Maintenance Staffing Problem
TARP	Tactical Aircraft maintenance Routing Problem
OARP	Operational Aircraft maintenance Routing Problem
FDARP	Flight Delay coupled with operational Aircraft maintenance Routing Problem
FSP	Flight Scheduling Problem
FAP	Fleet Assignment Problem
FAM	Fleet Assignment Models
ILP	Integer Linear Programming
FAA	Federal Aviation Administration
LOF	Lines of Flights
B&B	Branch and Bound
NPD	Non-Propagated Delay
PD	Propagated Delay
PMFs	Probability Mass Functions
CSP	Crew Scheduling Problem
CPP	Crew Pairing Problem
CRP	Crew Rostering Problem
MIP	Mixed Integer Programming
GT	Game Theory

LFSG	Leader-Follower Stackelberg Game
NG	Nash Game
MILP	Mixed Integer Linear Programming
CA	Compressed Annealing
ACO	Ant Colony Optimization
KTT	Karushe-Kuhne-Tucker
LFS	Stackelberg Game decision Model
NJOP	Non-Joint Optimization method
SNGM	Stackelberg-Nash Game Model
IATA	International Air Transport Association

Chapter 1 - Introduction

1.1 Research Background

Recently, aviation and airline maintenance providers are of the most significant worldwide industries. This is shown by the enormous growth in the number of passengers, which was around 3.5 billion passengers in 2015, expecting an annual growth of 5%[1]. To cope with this passenger growth, the number of aircraft is expected to increase from 24,579 in 2014 to 29,955 in 2022[2]. Consequently, the aircraft maintenance cost paid by airlines to maintenance providers is expected to increase from US $62.1 billion in 2014 to US $90 billion in 2024. Despite this pleasing economic situation for airlines and maintenance providers, many difficult challenges have been emerged during the planning and operating processes. One of the challenges facing airlines is how to build efficient routes for their aircraft, while respecting the operational maintenance restrictions. In this regard, aircraft maintenance routing problem (AMRP) is very significant for airlines due to its ability to construct the aircraft routes and schedule aircraft maintenance visits (Gopalan and Talluri, 1998, Liang et al., 2011). On the other hand, for the maintenance providers, it is a great challenge to manage the workforce capacity required to serve the increased number of aircraft. Therefore, maintenance staffing problem (MSP) is recognized as an effective tool for maintenance providers, as it manages the workforce capacity required to serve the airlines' aircraft (Yan et al., 2004).

[1] http://www.iata.org/about/Documents/iata-annual-review-2015.pdf
[2] https://www.iata.org/whatwedo/workgroups/Documents/ACC-2015-GVA/1630-1650-mtc-cost-trends.pdf

1.1.1 Aircraft Maintenance Routing Problem (AMRP)

AMRP, as an effective tool for airlines, is solved with the aim of building the route to be flown by each aircraft, while considering the operational maintenance restrictions. In the literature, AMRP has been discussed with three different scopes: tactical AMRP (TARP), operational AMRP (OARP), and flight delay coupled with operational AMRP (FDARP). Firstly, TARP focuses on generating cyclic rotations to be repeated by each aircraft (Gopalan and Talluri, 1998, Talluri, 1998). The drawback of TARP is overlooking some operational maintenance restrictions, which restricts implementing these rotations in real practice. Therefore, AMRP has been studied with an operational focus, in which the operational maintenance restrictions, such as the maximum flying hours, the maximum number of take offs, and one maintenance visit every four days, are considered (Haouari et al., 2012). Although OARP considers the operational maintenance restrictions, implementing its generated routes may not be viable due to ignorance the flight delays that frequently occur. This motivates researchers to combine flight delay and operational scopes, as in FDARP, in order to generate routes that better withstand disruptions (Liang et al., 2015).

Practically, airlines use FDARP to build efficient routes. Towards this goal, FDARP uses three things as an input, including the flight legs and their corresponding delay, the aircraft, and the maintenance providers, in order to generate an output, called the routing plan, as shown in Figure 1.1. This routing plan consists of group of routes, such that each route includes some maintenance visits. To implement the generated routing plan, two tasks must be performed. The first task is to cover the flight legs, which is the role of airline, whereas the second task is to handle the maintenance visits, which is the responsibility of maintenance providers. Practically, the role of maintenance providers is very important, as they are not only responsible for completing the maintenance operation, but also responsible for letting the aircraft depart from the maintenance station punctually. Therefore, the maintenance providers should efficiently manage their workforce capacity by solving the MSP. From this description, we can see that implementing routing plan is a mutual responsibility of the FDARP of airlines and the MSP of maintenance providers, and both problems are related interdependently.

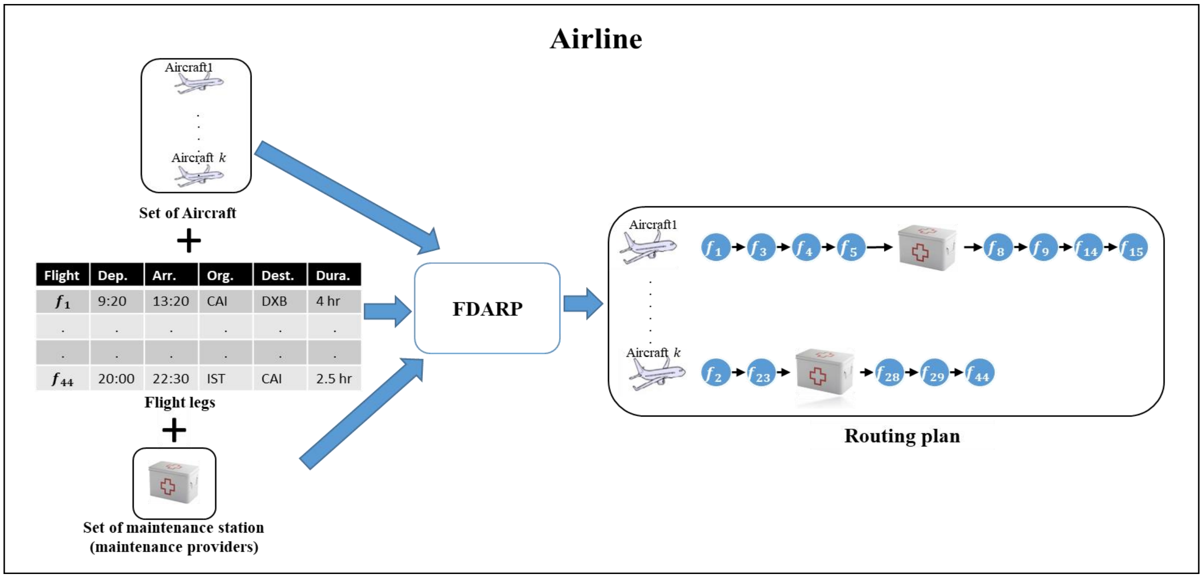


Figure 1.1: Representation of FDARP of airlines

1.1.2 Maintenance Staffing Problem (MSP)

Based on the maintenance demand obtained by solving the FDARP (e.g. the number of aircraft and their corresponding arrival and departure time), the MSP is solved so as to determine team sizes required to maintain the aircraft. For this purpose, the MSP uses the maintenance demand and the workforce capacity as an input in order to generate a so-called staffing plan, as shown in Figure 1.1. Practically, this plan includes groups of aircraft and their determined teams. Implementing the staffing plan necessities providing the teams, which is the role of maintenance providers, and sending the aircraft to the maintenance stations on time as the responsibility of airlines. If an aircraft fails to arrive on time, the staffing plan will be interrupted, as it might require more workers to complete the maintenance operation on time. So, implementing staffing plan is again a mutual responsibility of the MSP of maintenance providers and the FDARP of airlines, and both problems are related interdependently.

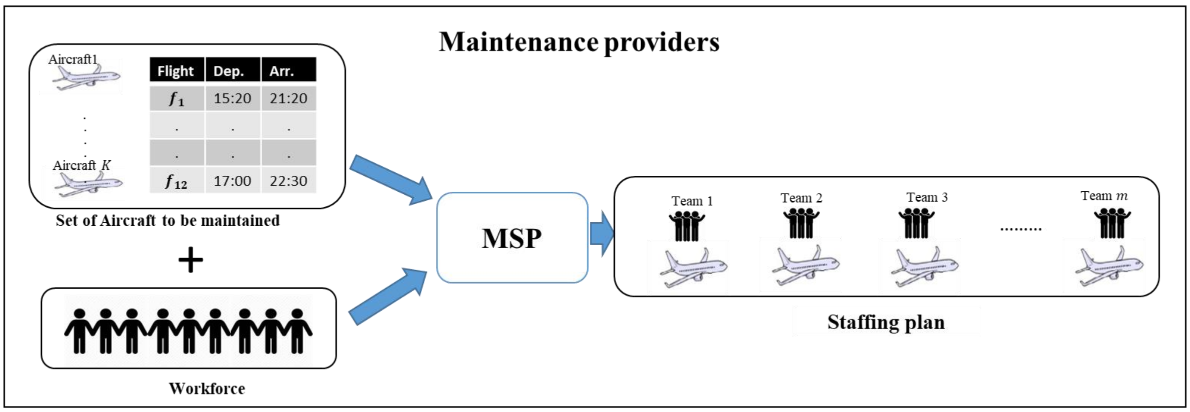


Figure 1.2: Representation of MSP of maintenance providers

1.2 Problem Statement

In the literature, aircraft maintenance routing problem and maintenance staffing problem have been extensively studied. However, there are still some research gaps that exist in the existing literature. These gaps can be identified as follows:

1 Most of the OARP models were formulated based on the set-partitioning formulation. This means that the number of feasible routes grows exponentially with the number of flight legs. Indeed, the weakness of this formulation is that it needs generating an exponential number of feasible routes, which result in disability of this formulation to handle large scale problems.

2 The majority of OARP studies considered some operational maintenance restrictions and neglected the rest. For instance, the restrictions regarding the maximum flying hours restrictions, the maximum number of take-offs, and one maintenance visits for every four days have been considered only on the models by Barnhart et al. (1998) and Haouari et al. (2012). However, the drawbacks of these studies include overlooking the working times and workforce capacity of maintenance stations, except the work by Haouari et al. (2012) that considered the workforce capacity of maintenance stations, but this consideration is relaxed in their computational experiments.

3 Most of FDARP studies are anchored on the expected value of the non-propagated delay. It should be noted here that the non-propagated delay happens due to some external factors, including airport congestion, passenger delays, and bad weather. The pitfall of the expected value approach is that the realized value of the delay for some flights turn out to be significantly different from their expected value, due to the high uncertainty of the delay.

4 One of the glaring facts in the literature is solving the FDARP of airlines and the MSP of maintenance providers independently, even though there is a clear interdependence between them. Therefore, the routing and staffing plans cannot be operated as planned.

5 Lastly, the price competition among the maintenance providers and its effect on the routing plan of airlines has not been investigated in the literature.

1.3 Research Scope and Objectives

This research mainly focuses on developing an OARP model that is able to handle real and large-scale AMRP problems. In addition, our focus is extended to develop a coordinated decision support system for FDARP and MSP that can efficiently handle the interrelationship between airlines and maintenance providers. Towards these goals, the research objectives can be summarized as follows:

- To develop a scalable OARP that can consider the workforce capacity and working hours of the maintenance stations, besides the other operational maintenance restrictions that are considered in the literature. In addition, to develop an efficient solution algorithm that can solve the developed AMRP model and generates high quality solutions in a short computational time.
- To develop a method that can better capture the non-propagated delay in real practice, and a coordinated decision support system for the FDARP of airlines and the MSP of maintenance providers, so that the interdependence between the two parties can be captured.
- To develop a coordinated decision support system that can capture the interdependence between FDARP of airlines and MSP of maintenance providers, in the presence of the price competition among the maintenance providers.

The main contribution of this research can be concluded from two different perspectives: academically, and practically. Firstly, from the academic perspective, this research fulfills the identified research gap in the area of aircraft maintenance routing problem and maintenance staffing problem, as mentioned earlier. Secondly, from the practical perspective, this research provides the aviation industry an efficient OARP model and a coordinated decision support system for FDARP and MSP, which helps in improving the existing routing and staffing plan and reducing their associated operational costs. The contribution of this study can be summarized as follows:

- Developing a scalable OARP model that adopts a polynomial number of decision variables and constraints. Consequently, the developed model can efficiently handle real and large-scale problems. Since this formulation is more scalable compared to the set-partition formulation, it considers the working hours and workforce capacity of maintenance stations, besides the other operational maintenance restrictions considered in the literature. To our best knowledge, this model is the first one that considers all these restrictions in a single model. In addition to this contribution, we propose an efficient solution algorithm that outperforms the existing solution methods, such as compressed annealing, in producing better solutions in a much shorter computational time.
- Developing a new scenario-based stochastic framework for FDARP in order to improve the representation of the non-propagated delay. In addition, we develop a coordinated decision support system by proposing a leader-follower Stackelberg game theoretic model, so that the interdependence between the new developed FDARP and MSP can be captured. Using such a model in real practice results in a significant saving for airline and maintenance providers, as it helps in operating the routing and staffing plan as planned.
- Developing a new Stackelberg-Nash game model, which consists of two sub-games: a Stackelberg game to capture the interdependence between FDARP and MSP, and a Nash game to capture the price competition among the maintenance providers. Moreover, we adopt data analytics by proposing a

neural network-based algorithm to accurately forecast the non-propagated delay. Using such a novel model practically, can reduce the costs of routing and staffing plans, and increase the profitability of the maintenance providers.

1.4 Research Significance and Value

It was shown that FDARP and MSP are connected closely with each other. Therefore, if one part is disrupted or fails to achieve its goal, it will induce a negative impact on the operation of the second part. The severity of that impact becomes great due to the rapid growth of air traffic and frequent happened disruption events. Therefore, the coordination between the two parts becomes necessary. By using the coordinated decision support model, the airlines and maintenance providers will get routing and staffing plans that can be operated as planned, resulting in a significant reduction of operational cost.

1.5 Structure of the Thesis

The rest of the thesis is arranged as follows:

- Chapter 2 presents a thorough review on the literature related to airline schedule planning, while focusing in two problems, called aircraft maintenance routing problem, maintenance staffing problem. In addition, game theory and some data analytics techniques are discussed. Then, in the end, the research gaps are summarized.
- Chapter 3 presents the developed OARP along with the proposed solution algorithm. Firstly, a modified connection network that represent the OARP is shown. Then, the mathematical model for the new developed OARP is presented.

In order to solve the proposed model, we describe the existing solution methods and their drawbacks, which motivates us to develop an efficient algorithm. To validate the potential of the proposed model and the solution algorithm, computational experiments are conducted based or real data acquired from a major airline located in the Middle East. These experiments are extended for two purposes. Firstly, to show the implications of the new considerations, like the workforce capacity. Secondly, to show the importance of the proposed solution algorithm, by making a comparison against existing solution methods, such as compresses annealing.

- Chapter 4 presents the leader-follower Stackelberg game model to capture the interdependence between FDARP and MSP. In this model FDARP acts as a leader, and MSP behaves as a follower. To solve the proposed model, we develop a bi-level ant colony optimization-based algorithm, which can find the Nash equilibrium for the game model. To demonstrate the potential of the proposed model, we present a case study for the proposed model while handling a real data acquired from a major airline and a maintenance provider located in the Middle East.
- Chapter 5, similar to Chapter 4, starts with the Stackelberg-Nash game model that reflects the interdependence between FDARP and MSP, in the presence of price competition. To find the overall Nash equilibrium for the proposed game model, we extended the developed model in Chapter 5 by combining it with an analytical model. To show the superiority and applicability of the proposed model, we extend the case study presented in the previous chapter, by collecting real data from one major airline and four maintenance providers located in the Middle East.

- Chapter 6 lists the conclusions made based on the results generated in this research. In addition, the existing limitations of the current research and future research directions are also pointed out by the end of this chapter.

Chapter 2 - Literature Review

In this chapter, we present the literature review for the planning processes carried out by two parties: airlines and maintenance providers. Firstly, for airlines, we discuss their planning process, which is called airline schedule planning, including flight scheduling, fleet assignment, aircraft maintenance routing, and crew scheduling. Also, we shortly review the integrated airline scheduling models, since these models have received attention from scholars in the last decade. Secondly, for maintenance providers, the maintenance staffing problem research work is discussed. In addition, this chapter presents a thorough review of the existing literature in the field of game theory and some tools of data analytics. Finally, by the end of this chapter, the research gaps are identified and elaborated.

2.1 Planning Processes of Airlines and Maintenance Providers.

Practically, airlines and maintenance providers have a close relationship. For better understanding this relationship, we discuss the planning process carried out by each part. Firstly, for airlines, their planning process can be called the airline schedule planning. This process consists of four stages, as shown in left-hand side of Figure 2.1. Usually, these problems are solved in an independent way or sequentially, in which each stage's solution forms the input for its subsequent stage. As we will see shortly, these four stages are not independent and partial or full integration of these stages produces better solution. It is out of question that it is possible to formulate the four stages in one model but solving that model will be intractable. Therefore, these stages are solved independently.

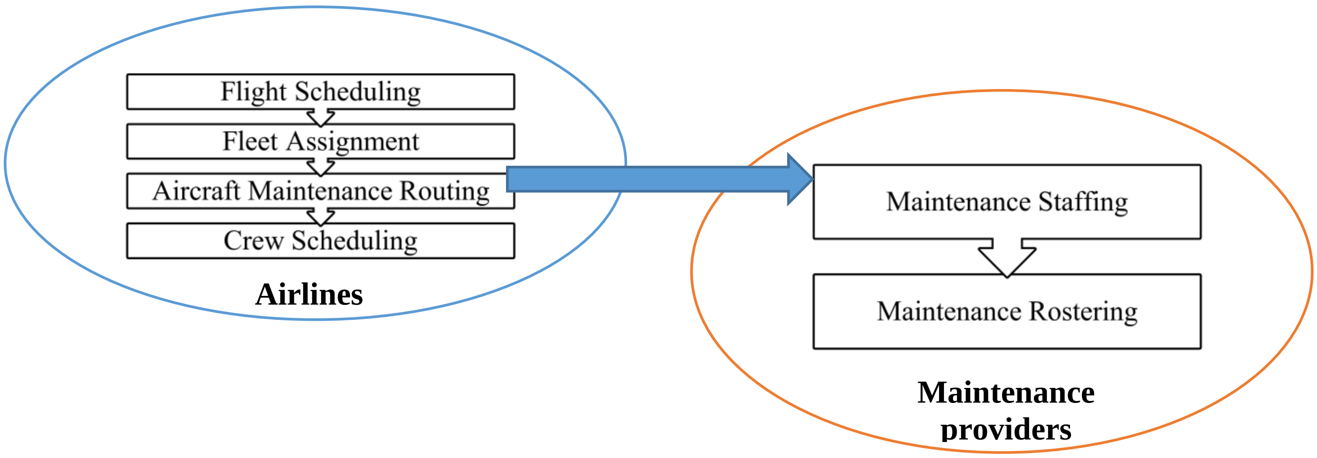


Figure 2.1: The planning processes carried out by airlines and maintenance providers

The left-hand side of Figure 2.1 elaborates the sequential operations carried out by airlines before the departure of the aircraft. These operations start by constructing the flight schedule, in which the passenger demand and ticket price are taken into consideration. Then, the airlines assign a specific fleet type to cover each scheduled flight by applying the fleet assignment. Thereafter, a feasible maintenance route is designed for each aircraft that belongs to a specific fleet by adopting the aircraft maintenance routing. These routes form the so-called routing plan for the airlines, which is used as an input for crew scheduling, in which cabin crew and cockpit are assigned to each flight. In addition, this routing plan acts as the linkage between the airline and the maintenance provider, as it is used as an input for the maintenance staffing, which specifies the team sizes required to maintain each received aircraft from airlines. These teams form the so-called staffing plan for the maintenance provider. Lastly, the staffing plan is used as an input for maintenance rostering to determine the working load for each worker. In the following sections, we review the research work done up to date regarding each stage and summarize the features of each model proposed in the literature.

2.2 Airline Schedule Planning

2.2.1 Flight Scheduling Problem (FSP)

The airlines start their schedule planning process by solving FSP with the aim of generating a timetable or a list of flights and their related information, such as the departure time, the origin, the arrival time, the destination. Towards this goal, the airlines consider the market conditions such as passenger demand and the ticket price. Usually, FSP is handled in two steps. Firstly, constructing the timetables. Then, evaluating this these timetables, by examining the feasibility of some flight legs.

The FSP studies can be categorized into two categories. Firstly, the models with market share and passenger demand consideration. Secondly, the models with robustness consideration.

2.2.1.1 FSP with passenger demand and market share considerations

Towards the goal of building an efficient timetable, it is important to consider two issues; the demand fluctuations and market share. Regarding the demand fluctuations consideration, it was appeared for the first time in the model by Yan and Young (1996), causing an improvement over the approach adopted by Taiwan airlines, called trial and error approach. Despite this improvement, this model overlooks one important aspect, named the market share consideration. The overlooked market share in the study by Yan and Young (1996) was considered in the study by Yan and Tseng (2002). The drawbacks of this model are twofold. Firstly, assumption of fixed market share, which was relaxed in the study by Yan et al. (2007). Secondly, assumption of fixed passenger demand, which was avoided in the study by Jiang and Barnhart (2009). All the previous studies did not consider the passenger demand and the market share simultaneously. These issues were considered on the study by Yan et al. (2008), in which the FSP was proposed as a stochastic programming model.

2.2.1.2 FSP with robustness consideration

Since airline is one of the industries that frequently faced by disruptions such as bad weather, technical problems, passenger delays, it is imperative to construct robust timetables that can better withstand these disruptions. This can be achieved in different ways. Firstly, by constructing a reliable timetable using retiming or adjusting the flight departure time (Lee et al., 2007). Secondly, by constructing flexible timetable, in which the airline mangers can find multiple swapping opportunities for the aircraft (Burke et al., 2010). Lastly, by developing a stochastic FSP model that considers the block time uncertainty (Sohoni et al., 2011). In a recent study by Jiang and Barnhart (2013), the robustness was presented in terms of the number of potential connected itinerary.

2.2.2 Fleet Assignment Problem (FAP)

Based on the constructed flight schedule, FAP plays its role in finding out which fleet will be picked to cover each scheduled flight leg. In the literature, FAP is formulated with two forms of objective functions. The first one is maximizing the profit, whereas the second one is minimizing the assignment costs. One of the challenges during this stage is finding the balance between the passenger demand and the aircraft capacities, as it has great impact on the airline profitability. For more clarification, imagine, for instance, we have two scenarios. The first scenario includes assignment of a large aircraft to cover a flight leg with small number of passengers, whereas the second scenario includes assignment of a small aircraft to cover a flight leg with large number of passengers. Both scenarios erode the profitability of the airline, such that in the first scenario, the seats are spoiled, and in the second scenario, the passengers are spilled. As a result, the accurate passenger demand should be considered while solving the FAP (Barnhart et al., 2002).

The next sections present the fleet assignment models (FAM) that appeared in the literature, including the basic FAM, and other models with different aspects such as variable departure times, network effect and robustness. In addition, we present the models with attention to the demand driven re-fleeting and the weekly planning horizon.

2.2.2.1 Basic FAM

This section presents the simplest form of FAM, which assumes that every day of the week has the same flight schedule. This can be called a daily planning horizon. The basic FAM was first discussed by Abara (1989) with the aim of finding feasible connections between two flight legs. Indeed, this formulation suffers from the enormous number of connections, which results in an increase in the model complexity. This problem was solved by Rushmeier and Kontogiorgis (1997) and Hane et al. (1995) who developed a preprocessing method that significantly reduces the number of feasible connections. Although the success of the basic FAM in finding which aircraft types that should be picked in order to fly the scheduled flight legs, some shortcomings were appeared, such as overlooking the network effect, the recapture of the spilled passengers, and variable departure times, which are discussed in the following sections.

2.2.2.2 FAM with variable departure time

There is one main advantage behind considering variable departure time for flight legs. This advantage is offering the flexibility for the FAM. The variable departure time was firstly considered in the study by Levin (1971). However, aircraft capacity and multiple fleet types were not considered in the proposed model. The work by Levin was extended by Desaulniers et al. (1997b) who developed a model in which the variable departure time and heterogeneous fleet were considered simultaneously. On an attempt that is similar to Levin's model, Rexing et al. (2000) proposed a model with variable departure time for each flight leg. This variability was carried out by providing a time window in which the departure time can vary, which in turn results in increasing the flight connection opportunities.

2.2.2.3 FAM with network effect

The network effect or flight leg interdependency means that the availability of seats on the flight legs directly affects the demand that comes from multi-leg passengers. The Ph.D. dissertation by Farkas (1995) was the pioneer work to address the network effect. This work was further extended by Barnhart et al. (2002) who considered the recapture of a spilled passenger that was neglected in the Farkas's work. The previous two research works did not pay attention to the stochastic passsneger demand as it is affected by the avaialility of seats on the flight legs. This point was considered by Jacobs et al. (2008) who used demand uncertainty to capture the stochastic demand. The stochastic passenger demand was also discussed by Dumas et al. (2009) who proposed a model that improved the profitability of Air Canada

2.2.2.4 Robust FAM

Indeed, few robust FAM were reported in the literature. In particular, Rosenberger et al. (2004) proposed the concept of isolate hubs for their robust FAM models. This concept means generating many short cycles that is less sensitive to flight cancelation. Robust FAM was also reported in the work by Smith and Johnson (2006) who presented a model, in which the station purity concept was incorporated. This concept means restricting the number of fleets that can serve each station.

2.2.2.5 Demand driven re-fleeting FAM

This section discusses one important tool that is applied by airlines to mitigate the demand uncertainty. This tool is called demand driven re-fleeting. The main concept behind this tool is updating the demand information, then using these updates to change the initial fleeting. This tool was first appeared in the work by Berge and Hopperstad (1993) who showed 1-5% as an increase in the profit after using this tool, while using US domestic carrier data. Talluri (1996) extended Berge and Hopperstad's work through proposing an efficient heuristic that can provide a solution to the model in a reasonable short computational time.

Jarrah et al. (2000) presented another re-fleeting model, which allows the managers to construct different re-fleeting scenarios and select the best re-fleeting scenario. Sherali et al. (2005) attempted to present a scalable model for the demand re-fleeting model presented by previous studies. For this purpose, the authors proposed re-fleeting model based on a polyhedral structure. This structure showed good performance while making a dynamic reassignment and demonstrated better demand forecasts.

2.2.2.6 FAM with weekly planning horizon

In the previous sections, the FAM models assume that every day of the week has the same flight schedule, except the work by Berge and Hopperstad (1993) and (Dumas et al., 2009). Practically, this assumption is not viable, as airlines permit different daily flight schedules. This flight schedule variation is a response from airlines to the demand fluctuation of different flights, as the demand at weekend of some flights is higher than on other days. To cope with this situation, FAM with weekly planning horizon is more applicable (Bélanger et al., 2006, Pilla et al., 2012).

2.2.3 Aircraft Maintenance Routing Problem (AMRP)

After solving the FSP and FAP, as in the previous two stages, now it is the turn for AMRP to start its role. Indeed, AMRP aims to determine the route to be flown by each aircraft in the fleet, while considering some operational maintenance restrictions, such as maximum flying hours, maximum number of take-offs, and one maintenance visit every four days. Before violating these operational maintenance restrictions, each aircraft should visit the maintenance provider to receive the required maintenance check, as mandated by federal Aviation Administration (FAA). Practically, there are four maintenance checks, which their frequency is different. For example, Type A maintenance check should be performed when the aircraft complete an accumulated 65 flying hours, whereas Type B maintenance check should be performed every 300-600 flying hours. The previous two checks are short-term checks. The long-term checks include Type C and Type D, which are performed every one and four years, respectively. Practically, airlines are operated based on some rules that are more stringent compared to those mandated by FAA. For example, instead of performing Type A maintenance check every 65 flying hours, it is carried out every 35-40 flying hours. Since Type A maintenance check is the most frequent check among the others, it is more considered by AMRP than others. In the next sections, we discuss the AMRP research work reported in the literature with more details, as this problem is the focus of this thesis. By reviewing the AMRP related studies, we can categorize AMRP into three main categories: tactical AMRP (TARP), operational AMRP (OARP), and flight delay coupled with operational AMRP (FDARP). These categories are discussed in the following sections.

2.2.3.1 Tactical AMRP

TARP studies focus on generating cyclic rotations to be repeated by each aircraft. The TARP was first appeared in the study by Kabbani and Patty (1992) that proposed a set-partitioning formulation for AMRP model. This model aimed at constructing feasible routes or lies of flights (LOF) for AMRP with 3-day planning horizon. In order to find LOF for AMRP with k-days planning horizon, Gopalan and Talluri (1998) presented a novel polynomial time-based algorithm, so that the LOF can be achieved for the static and dynamic formulations of the AMRP. Talluri (1998) extended the work by Gopalan and Talluri and proposed an effective heuristic in order to solve AMRP with 4-day planning horizon. Clarke et al. (1997) formulated AMRP as an asymmetric travelling salesman problem with the objective of maximizing the through value. It should be noted that the through value can be defined as the additional profit gained from the passengers attracted by the flight connections with short transit times. To find out these connections, the authors applied Lagrangian relaxation as a solution method. More recently, the daily AMRP was studied by Liang et al. (2011) who developed a novel network, called rotation-tour-time-space network. This network constituted the basis for proposing an integer linear programming (ILP) model, which was solved using a commercial software, called CPLEX. We can see that the previous studies can generate aircraft routes, which however may not be viable to be applied in real practice for two reasons. Firstly, these routes overlook some of the operational maintenance restrictions. Secondly, it is difficult to repeat these routes as airlines tend to change the flight schedule of each day of the week to cope with the passenger demand fluctuations. This motivates researchers to study AMRP with more operational focus, as in the next section.

2.2.3.2 Operational AMRP

OARP aims to determine aircraft routes, while respecting the operational maintenance restrictions, such as maximum flying hours, maximum number of take offs, and one maintenance visit every four days. These restrictions are mandated by FAA. In addition, it is important to consider the workforce capacity of the maintenance stations in order to avoid visiting maintenance stations with insufficient workforce capacity. Studying OARP received less attention from researcher than TARP. For example, Sriram and Haghani (2003) developed an effective heuristic to solve their proposed OARP model in which one maintenance visit every four days is considered as an operational maintenance restriction. In addition, the authors also considered the workforce capacity of maintenance stations. Moreover, The proposed model was extended by the authors in order to consider the maximum flying times as an additional operational maintenance restriction, but the authors failed to solve it. Sarac et al. (2006) studied OARP by proposing a set-partitioning formulation for OARP, then adopted branch-and-price to solve their proposed model. The authors considered in their model one operational maintenance restriction; the maximum number of flying hours. In a more recent study, Haouari et al. (2012) proposed a non-linear formulation for OARP, in which all the three operational maintenance restrictions were respected. It is worth mentioning that Haouari’s work considered the workforce capacity of maintenance stations, but this consideration was relaxed in the computational experiments. Towards the goal of solving the proposed model, the authors linearized the non-linear formulation by adoption of a reformulation-linearization technique. Then, they solved the linearized model using commercial software, called CPLEX. Başdere and Bilge (2014) proposed an ILP model for OARP that respected the maximum number of flying hours as an operational maintenance restriction. To solve the proposed model, branch and bound (B&B) and compressed annealing were adopted to handle small and large-scale problems, respectively. Generally, in reality, the sudden changes happen frequently, such as machine breakdown in production scheduling (Cai and Zhou, 1999, Cai et al., 2004, Cai et al., 2009b), the flight delays in aviation industry (Yen and Birge, 2006), port congestion in liner shipping (Li et al., 2016), and filters in control systems (Hashim et al., 2018b, Hashim et al., 2018c, Hashim et al., 2018a). Using OARP is useful in generating routes that satisfy the operational maintenance restrictions. However, implementing these routes is a matter of question, as these routes ignore one of the important and frequently happen factor that is abovementioned, called flight delays. This results in constructing routes that are sensitive to disruptions. Towards the goal of generating routes that better withstand disruptions, the operational AMRP was coupled with flight delays, as appeared in the last category of AMRP.

2.2.3.3 Flight delay coupled with operational AMRP

The FDARP aims to construct aircraft routes that respect all mentioned operational maintenance restrictions besides considering the flight delays. In the literature, few FDARP studies were reported. Lan et al. (2006) were among the first researcher to consider flight delay. The authors developed an FDARP model with the objective of minimizing the expected propagated delay. To do so, the authors developed a retiming approach, in which the flight departure times can be adjusted, so that the propagated delay can be significantly absorbed. This retiming approach was also used by Dunbar et al. (2014), but the authors incorporated the stochastic delay information in their model for better calculation of the expected propagated delay. Inserting time buffers between flight legs is another approach for mitigating the propagated delay, as appeared in the study by Liang et al. (2015).

The flight delay considered by FDARP can be categorized as a propagated delay (PD) and a non-propagated delay (NPD). For the NPD, it ascribed to bad weather, maintenance station congestion, technical problems, peak seasons, and passengers, which are generalized as non-routing reasons. On the other hand, the PD is described as any delay occurs when the aircraft covering a later flight is delayed due to a delay on its previous covered flight. Generally, the role of FDARP is to minimize the propagated delay after forecasting the non-propagated delays. In the literature, the non-propagated delay is forecasted by using expected value approach, which only focuses on analyzing the historical flight delay data. For example, Liang et al. (2015) collected the NPD for the top three fleets with the longest average PD and constructed the probability mass functions (PMFs) of NPD for each fleet. Next, they constructed a single non-propagated delay PMF by taking the average of the constructed three PMFs and used the single PMF to calculate the expected NPD.

2.2.4 Crew Scheduling Problem (CSP)

CSP is considered the last airline schedule planning stage, in which all the previous stages are assumed to have been solved. In the literature, CSP received much attention from scholars if compared with the other stages. This is because CSP manages the crews whose costs come in the second rank after the fuel cost, so any optimization in the crew cost can result in a significant profit improvement for airlines. Usually, CSP is solved with the aim of assigning a crew member for each scheduled flight leg. To do so, two problems should be solved, including crew pairing problem (CPP), and crew rostering problem (CRP). Firstly, CPP is solved to determine a set of pairing (crew members) that should cover a set of flight legs exactly once, while considering some crew regulations and contract issues. Secondly, CRP is solved using the solution generated by CPP in order to determine the working load or roster for each crew member, while respecting some working regulations like the total actual flying time and the number daily pairing, and others. In the following sections, CPP and CRP are discussed. In addition, the integration between these two problems is also presented.

2.2.4.1 Crew Pairing Problem (CPP)

As mentioned earlier, CPP is solved with the aim of constructing an anonymous pairing so that each scheduled flight leg is covered by exactly one pairing. By reviewing the CPP, we can divide its related work into four types: basic crew pairing models, crew pairing models that are implemented by airlines, stochastic crew pairing model, and lastly the robust crew pairing models.

The basic crew pairing models are developed while neglecting the uncertainty issues occurred in the airline industry. These models can be categorized into two categories: models with daily planning horizon, and models with weekly planning horizon. For daily planning horizon models, it is assumed that that every day of the week has the same flight schedule (Hoffman and Padberg, 1993, Barnhart et al., 1995, Vance et al., 1997). For the weekly planning horizon models, on the other hand, different daily flight schedules are assumed (Desaulniers et al., 1997a, Klabjan et al., 2001, Yan and Chang, 2002).

In the literature, there are some research work that succeeded in improving the existing airline system, resulting in implementation of these models in real practice. For example, US Airline adopted the SPIRIT system developed by Anbil et al. (1991) as it cause US$ 20 million as an annual saving, while constructing the crew pairing. United Airlines also implemented the system proposed by Graves et al. (1993) and yielded an annual saving for about US$ 16 million, while handling medium and large-scale crew problems. In addition, Chu et al. (1997) proposed a successful crew pairing model, which motivated American Airlines to implement it in their crew planning stage, resulting in an annual saving of US$ 2 million.

All the previous crew pairing models are modeled as deterministic models, while overlooking the uncertainty issue such as weather changes, and airport congestion. This result in generating crew pairing that can be easily disrupted by these issues. For this purpose, stochastic crew pairing problem was emerged. Few stochastic crew pairing studied are reported. For example, Schaefer et al. (2005) were among the first to develop a stochastic crew pairing model, which showed good practical performance during the frictional disruptions (weather changes, and airport congestion) if compared with the deterministic models. Yen and Birge (2006) extended the work by Schaefer et al. (2005) by proposing a two-stage stochastic model, which showed a significant cost savings, while considering the uncertainty in the crew pairing stage.

Since airlines are frequently faced by sudden circumstances like technical problems and airport congestion, it is necessary to develop robust crew pairing models, which provide crew pairing that can better withstand these disruptions. This motivated scholars to study robust crew pairing models. For example, Ehrgott and Ryan (2002) proposed a robust model, in which the robustness is captured by penalizing any connection with short time as it might result in the delay. Shebalov and Klabjan (2006) also proposed the idea of maximizing the move-up crews or swapped crews as a variant for robustness. This idea simply means increasing the number of crew swapping opportunities so that it will be easy to avoid the disruptions. Moreover, Tekiner et al. (2009) developed a robust model, in which the swapping option to manage the extra flights was adopted. The proposed model can only solve the small-sized problem. To handle large-sized problems, Muter et al. (2013) further extended the work by Tekiner et al. (2009) and developed a solution method based on column generation approach.

2.2.4.2 Crew Rostering Problem (CRP)

After generating the crew paring, CRP starts its role by finding out the working load or roster for each crew member, while considering vacations, crew requests, skills, regulations, company rules, and union agreements. This step includes generating the crew schedule in which each flight is covered by the required crew attendants and cockpit members. Practically, CRP is mainly solved with the objective of minimizing the operational cost, aiming at constructing a monthly crew schedule. Towards this goal, there are two methods that can be used; the rostering system and the preferential bidding system.

The rostering system is usually implemented by European airlines like Swiss Air, Air France, and Alitalia. The main concept behind this system is to construct individual rosters for crew members by considering the crew requests and the pre-assignments activities. In addition, this system tries to construct fair and equal share crew schedule. This system received much attention from researchers, as shown by (Ryan, 1992), (Day and Ryan, 1997), (Gamache et al., 1999), (Lučic and Teodorovic, 1999), (Dawid et al., 2001), (Cappanera and Gallo, 2004), (Maenhout and Vanhoucke, 2010), and (Fahle et al., 2002).

The preferential bidding system is another method to construct the crew schedule. Practically, this system is commonly used by US and Canadian airlines. This system is similar to the rostering system as both consider the pre-assignments activities, but the preferential bidding system is unique in considering the crew member preference by using weighted bids. While many research works focused on the rostering system, few research works discussed the preferential bidding system, as shown by (Gamache et al., 1998), and (Achour et al., 2007).

2.2.4.3 Integrated crew scheduling problem

In the previous section, we present CPP and CRP, which are solved independently. However, the main drawback of the independent approach is the sub-optimality solutions, meaning that the optimal solution in one stage is not more optimal in subsequent stages. Towards the goal of avoiding the sub-optimality solutions, researchers changed their interest from solving both problems independently, and focused on the partial and the full integration of both problems.

Partial integration of CPP and CRP means solving both in a sequential approach, as shown in the work by Guo et al. (2006). The authors proposed a partial integrated crew scheduling model that caused a significant reduction in the crew cost, if with the independent approach. For the full integration of CPP and CRP, on the other hand, it handles both problems in a simultaneous approach. This approach was reported by more scholars if compared with the partial integration approach, as shown by (Zeghal et al., 2011), (Souai and Teghem, 2009), and (Saddoune et al., 2011).

2.2.5 Integrated Airline Schedule Planning Models

While many researchers paid attention on solving a single stage as in the previous sections, other researchers focused on solving the integrated stages to escape from the sub-optimality issue. In this section, we review AMRP while integrated with other stages, as our focus in this research is on AMRP.

2.2.5.1 AMRP integrated with FAP

AMRP was integrated with FAP with the aim of assigning aircraft type or fleet for each flight leg, while satisfying the operational maintenance restrictions, mandated by FAA. This integration was appeared in some studies. For example, Haouari et al. (2009) proposed an efficient heuristic as a solution method, which could find near optimal solutions for real data from Tunis Air. Zeghal et al. (2011) proposed a model that incorporated the idea of aircraft renting. This idea means renting out and hiring the aircraft during the low demand and high demand seasons, respectively. This idea caused US$ 33.8 million as an improvement in annual profit, while using real life data from Tunis Air. In the study by Haouari et al. (2011), the short term and long term maintenance issue were presented. Liang and Chaovalitwongse (2012) studied the integrated FAP and AMRP, but for weekly planning horizon. For this purpose, they developed an innovative network to capture the discussed problem.

2.2.5.2 AMRP integrated with CPP

For joint aircraft routing and crew scheduling, Cordeau et al. (2001) developed an integrated model in which the maintenance issues and minimum connection time were considered. This model was solved by using two solution methods: bender decomposition and column generation. The proposed model caused reduction in the crew cost by about 9.45% over the approach that solves both problems independently. Mercier et al. (2005) extended Cordeau et al.'s model by proposing a robust model, in which the connections that may cause delay were penalized. This showed an improvement in the solution robustness. Providing a robust model also received attention from the work by Weide et al.

(2010). The authors used an iterative approach, which includes solving crew problem then using the crew solutions to solve the routing problem. This process keeps iteration until there is no more chance to improve the solution robustness.

Dunbar et al. (2014) studied the delay propagated issue by inserting information of stochastic delay in their proposed model. To minimize the propagated delay, the authors used two algorithms, which include by re-timing the departure time of scheduled flight legs. The proposed model showed a good practical performance as it caused a reduction on the propagated delay by 14%. Díaz-Ramírez et al. (2014) proposed full integration and partial integration models for AMRP and CPP, which considered a single fleet and a single maintenance and crew base.

All the previous integrated AMRP and CPP model were integrated using the full integration approach, meaning that each problem's decision variables are included in the objective function. However, some researchers focused on studying the integrated AMRP and CPP using the partial integration approach, as shown in the work by Klabjan et al. (2002) and Cohn and Barnhart (2003).

2.2.5.3 Integration of AMRP with two different planning problems

In a research attempt that aimed to integrate AMRP besides FSP and CPP, Mercier and Soumis (2007) developed a model that allows adjusting the departure time of flight legs. The proposed model was solved by using benders decomposition method.

On the focus of integrating AMRP with FSP and FAP, Sherali et al. (2013) proposed a model in which many issues were taken into consideration, including departure time re-timing, multiple fare classes, demand recapture, through flights, and maintenance issues. To solve the proposed model, the authors adopted the benders decomposition approach. The computational results of the model demonstrated a better profitability when compared with the method that solved each problem independently. In addition, Gürkan et al. (2016) discussed for the first time the cruise speed and incorporated in their model, so that the fuel utilization is increased and the number of needed aircraft is decreased.

The integration of AMRP with other two phases, called FAP and CPP, was noticed in some studied in the literature. For example, Cacchiani and Salazar-González (2013) proposed a model and solution algorithm with the aim of solving their integrated model. Towards the goal of validating the developed model, computational experiments were conducted based on real data from a Canary island carrier. The results showed that the proposed model outperformed the manual solution method applied by the Canary island carrier. Also, Salazar-González (2014) proposed a scalable model that integrated FAP, AMRP, and CPP. The proposed model was applied to plan the daily flights operated by a Spanish funded project. Moreover, Shao et al. (2015) developed an innovative strategies that accelerated the bender decomposition, which was adopted as a solution methodology. In a recent study by Cacchiani and Salazar-González (2016), two solution methods were proposed in order to solve the integrated model. These methods called path-path and arc path methods. To demonstrate the efficiency of the proposed methods, a real test cases was used in the computational experiments. The results showed an outperformance for the second method over the first method, while looking for optimal solutions.

2.3 Planning Process of Maintenance Providers

2.3.1 Maintenance Staffing Problem (MSP)

MSP is considered one of the main planning processes carried out by maintenance providers. This stage is conducted by using the maintenance demand obtained from the AMRP solution (e.g. the number of aircraft and their corresponding arrival and departure times) as an input. Based on the received maintenance demand, MSP is solved so as to determine the team sized required to maintain the aircraft, with an objective of minimizing the total operating cost. It is necessary for the maintenance workforce staffing plan to meet the required demand, as well as to allow the flight arrival and departure to remain punctual. This problem is the second focus of this research.

The aviation safety is carefully regulated, and the maintenance operations should be performed according to these regulations. Such maintenance to be completed requires many skilled technicians and advanced instruments. Therefore, maintenance work is separated into three different types. First type is called short-term maintenance plan, and it usually covers the line maintenance that are performed at the airport gates. Second type is called the mid-term maintenance plan and covers Type A and Type B maintenance checks. To perform mid-term maintenance plan, it is necessary for the aircraft to move to the hanger located in the airport to complete the maintenance check. Long-term maintenance plan is considered the last type that covers type C and D maintenance checks. This type needs the aircraft to go out of services and stay at a separate location while performing the maintenance.

MSP is one of the problems that did not receive much attention from scholars in the literature. Dietz and Rosenshine (1997) studied MSP by developing an approach that could determine the optimal sizes of the workforce teams that are required to perform maintenance for military aircraft. Beaumont (1997) also presented a mixed integer programming (MIP) model for MSP, with the objective of minimizing the manpower supply. Using the same objective by Beaumont's work, Yang et al. (2003) developed a MIP model that aimed at determining the team sizes, besides finding out the number of day shifts and their related time. the proposed model considered different flexible management strategies in order to manage the manpower supply in an efficient way. In a follow up paper, Yan et al. (2004) expanded their previous work by considering some certification constraints, such as the training levels of the workforce members and the degree of functional abilities. The model viability was tested based on a case study from Taiwan airlines.

2.4 Game Theory

In this research study, game theory (GT) has been used as a tool to manage the interrelationship between airlines and maintenance providers. GT studies the strategic interaction among different decisionmakers who behave rationally to make decisions that potentially affect the interest of other decisionmakers. This study focuses on two forms of GT, including the leader-follower Stackelberg game (LFSG), and the Nash game (NG).

The first form adopted in this study is LFSG, which reflects the coordination between two players who have conflicting goals, known as the leader and the follower (Yang et al., 2015). This game is initiated by the leader who firstly makes the decisions, as holding the dominating position in the game, and passes these decisions to the second player, known as the follower. The follower, as holding the dominated position in the game, reacts rationally by using the received decisions by leader and makes the decisions, which are sent back to the leader. This form of the game has shown successful application in many areas, such as the seller-buyer supply chain (Esmaeili et al., 2009, Xiao et al., 2014), product families and supply chain (Yang et al., 2015, Wang et al., 2016), inventory policies in vendor managed inventory (Yu et al., 2009), and pricing (van Hoesel, 2008).

NG is considered the second game used in this research study. This form of the game is frequently used in the literature with the aim of capturing the competition among different players, while setting the price of their product or service (Yu and Huang, 2010). The process of the price competition is proceeded as follows. It starts when each player sets a price for his product or service, which is later evaluated by the customer in the form of the product or service demand. This price is observed by the other players or competitors in the market who in turn react rationally towards this price and adjust the price of their products or services with the aim of attracting more demand in order to finally maximize their profit. By looking precisely to this process, we can see each player sets the price for his product or service by considering not only his prices but also the prices offered by his competitors in the market. Similar to LFSG, NG has been successfully applied in different fields, such as manufacturer-retailer supply chain (Choi, 1991, Sinha and Sarmah, 2010), pricing in supply chain (Cai et al., 2009a), advertising in vendor managed inventory (Yu and Huang, 2010), and rail transportation system (Hsu et al., 2010).

2.5 Data Analytics

Data analytics is another tool adopted in this research study. In the literature, Data analytics can be defined as the application of various tools, such as data mining, statistical tools, etc. on the data with the objective of examining and analyzing the data to finally discover correlations, trends and other valuable information (Tiwari et al., 2018, Ghofrani et al., 2018). Among various data analytics tools, neural network and regression algorithms are of most efficient tools to capture the relationship between a response variable and one or multiple predictors. The efficiency of the neural network was reported in different fields, such as demand forecasting (Tsai et al., 2009), liquidity risk assessment in banking (Tavana et al., 2018), and prediction of organ status in healthcare industry (Misiunas et al., 2016). On the other hand, the regression algorithm showed successful applications, including delay and demand forecasting in railroad industry (Murali et al., 2010, Batley et al., 2011), price prediction in warm-water fish supply chain (Tabrizi et al., 2017), and acceleration prediction for railway wagons (Shafiullah et al., 2010).

2.6 Discussion

The literature reviewed above revealed the following research gaps:

1 The set-partitioning formulation has been used for modeling OARP. It is commonly known that to use this formulation, it is necessary to generate all the feasible routes, which grows exponentially with the number of flight legs (Sarac et al., 2006). The pitfall of this formulation is that it results in producing an exponential number of feasible routes, leading finally to the disability of this formulation to handle large scale problems. This observation is reported in the

work by Sarac et al. (2006), in which the proposed model failed to handle large scale problems, and only handled small scale problems with 175 flight legs.

2 It is observed in the most of OARP that some operational maintenance restrictions were considered, whereas the rest of the restrictions are overlooked. For instance, the restrictions regarding the maximum flying hours restrictions, the maximum number of take-offs, and one maintenance visits for every four days have been considered only on the models by Barnhart et al. (1998) and Haouari et al. (2012). However, the drawback of these studies is overlooking the working times and workforce capacity of maintenance stations, except the work by Haouari et al. (2012) that considered the workforce capacity of maintenance stations, but this consideration is relaxed in their computational experiments. Neglecting the workforce capacity might result in assigning the aircraft to a maintenance station that do not sufficient workforce capacity, leading to a prolonged maintenance duration. This long duration can be avoided by calling more hands, which in turn results in additional cost incurred for overlooking the maintenance stations workforce capacity. Meanwhile, if the working hours of maintenance station is neglected, it is highly probable for aircraft to arrive the maintenance station at times outside its working hours. This results in a delay for the aircraft, which in turn causes a cancellation for the subsequent scheduled flights.

3 Most of FDARP studies are formulated using the approach of expected value of the non-propagated delay. It should be noted that the non-propagated delay happens due to some external factors, such as technical problems, airport congestion, bad weather, and others. The pitfall of the expected value approach is

that the realized value of the delay for some flights turn out to be significantly different from their expected value, due to the high uncertainty of the delay. As a result, the flight delays are propagated, and their related cost is maximized.

4 Although the FDARP of airlines and the MSP of maintenance providers have been extensively studied in the literature, the interdependence existed among both problems has not been investigated in the literature. In fact, both problems are closely interrelated. For airlines, the routing plan can be easily interrupted if the aircraft cannot be released from the maintenance station punctually. Similarly, for maintenance providers, the staffing plan can be easily interrupted if the aircraft missed the scheduled maintenance visit. So, if both problems are solved independently, the staffing and routing plans cannot be operated as planned, resulting in a severe flight delays for airlines and huge interruptions for the maintenance providers.

5 Recently, maintenance providers have been serving in a competitive market, in which they are struggling to survive. One of the ways to survive is attracting more maintenance demand from your competitors. To do so, the maintenance providers use the price competition process, which includes cutting down the price of the maintenance service. Observing this action by airlines results in changing their scheduled maintenance visits to target the provider with cheaper maintenance service. This can easily interrupt the routing plan of airlines. So, besides the interdependence existed among the FDARP of airlines and the MSP of maintenance providers, there is another factor that can easily affect the routing plan. In the literature, there is no research work pertaining to discuss the described

interdependence, in the presence of price competition among maintenance providers.

In order to fulfill the research gaps, three models are proposed, and their related computational experiments are conducted in this research work. The first model is designed to cover the first two research gaps. The third and fourth research gap are covered by employing the second model. Lastly, we design our third model to fulfill the last research gap. Detailed description, formulation, and solution methodologies for each model are introduced in the following chapters.

Chapter 3 - Model with a Solution Algorithm for the Operational Aircraft Maintenance Routing

3.1 Introduction

In this chapter, our focus is on OARP, and our aim is twofold. The first aim is to develop a scalable OARP model, in which a polynomial number of decisions variables and constraints are used. This polynomial feature enables us to consider all the operational maintenance restrictions in a single model. To achieve our first aim, we develop a new MILP model for OARP. The second aim is to solve real and large-scale test instances with a proposed efficient solution algorithm. To validate the performance of the proposed solution algorithm, we conduct computational experiments based on real data obtained from a major Middle Eastern airline. These experiments include comparisons between the results obtained by the proposed algorithm and the optimal results obtained by CPLEX, while tackling small-scale test instances. For large-scale test instances, to evaluate the performance of the proposed algorithm, the best upper bound is adopted. To show the importance of the proposed solution algorithm, we make a comparison between our algorithm and compressed annealing, as being one of the efficient existing algorithms. To demonstrate the implications of the new considerations, we modify the proposed OARP in order to be like the models in the literature. Next, a comparison between our model and the modified model is conducted, in order to see the implications of these considerations on the model, especially, the workforce capacity restrictions.

The rest of this chapter is organized as follows. Section 3.2 presents the model description, whereas the mathematical model formulation is described in section 3.3. To solve the proposed model, we propose an effective solution algorithm in section 3.4. In section 3.5, the comparison between the proposed OARP model and the existing models in the literature is discussed. In section 3.6, with the objective to validate the proposed model, computational experiments based on real data are provided. Lastly, a summary of this chapter is provided in section 3.7.

3.2 Model Description

By using three types of data sets: flight legs, aircraft, and maintenance stations, as an input, our proposed OARP is solved with the aim of building maintenance feasible routes for the aircraft, called the routing plan. In the literature, a route can be considered a maintenance feasible route if it respects three main operational maintenance restrictions, including maximum flying hours, maximum number of take-offs, and one maintenance visit every four days. However, these routes neglect two important restrictions, called the workforce capacity and working hours of maintenance stations. Imagine that the model overlooks the workforce capacity of maintenance stations and assigns three aircraft to perform maintenance in a station that suffers from a capacity shortage. In this situation, there is high a probability for the maintenance duration to be prolonged. This long maintenance duration can be avoided in the case of deploying more hands, which incurs extra costs due to overlooking the limits of the workforce capacity. For the maintenance working hours, such as the time of opening and closing, if this consideration is neglected, it will result in aircraft arriving at times that are different than the maintenance working hours, leading finally in prolonging the waiting time for the aircraft at the airport till the maintenance stations are operational. As a result, the next flights to be covered by the aircraft are cancelled, resulting in an additional cost paid to recover these cancelled flights. The previous description reveals the importance of considering the maintenance workforce capacity and maintenance working hours. Therefore, it is necessary to add these considerations on our proposed OARP. As mentioned earlier, our model formulation is scalable as being a polynomial formulation, therefore, in a single model, we can consider all the operational maintenance restrictions besides the workforce capacity and working hours of maintenance stations. These features ease implementing the model in reality.

In our proposed model, the objective is to maximize the profit. The profit is calculated as the difference between the through value (revenue) and the total penalty cost. For the through value, it can be defined as the collected revenue from through connects, which are the consecutive flights with short transit times that attract more passengers. For the penalty cost, on the other hand, it is the cost paid by airlines to maintenance providers in the case of violation the workforce capacity of maintenance stations. This cost can be considered as a compensation to call in more hands to complete the maintenance operations for the excess aircraft without significant delay.

3.3 Mathematical Model Formulation

3.3.1 Scope and notations

Before presenting the mathematical formulation of the model, we first define the scope of the model as follows:

- A 4-day planning horizon. We select this planning horizon due to some reasons. Firstly, in the literature, this planning horizon is one of the most used planning horizons (Feo and Bard, 1989, Talluri, 1998). Secondly, in real practice, to ease satisfying the operational maintenance restriction of one maintenance visit every 4 days, airlines prefer to repeat every 4 days the same flight schedule. In the light of these reasons, we select the 4-day planning horizon to be adopted in our proposed model.
- The existing maintenance stations are only considered in our proposed model, and there are no suggestions for building new stations.
- The hub airports host the maintenance stations.
- The number of workforce teams in each maintenance station is characterized as a deterministic variable.
- The proposed model only considers Type A maintenance checks, as it is the most frequent one among the others.

After presenting the scope of the proposed OARP model, we define the notations used throughout the model as follows:

Sets

NF:	Set of flight legs indexed by i and j.
MT:	Set of maintenance stations indexed by m.
K:	Set of aircraft indexed by k.
A:	Set of airports indexed by a.

$v \in (1,2, \dots, V)$: Average number of maintenance operations that each aircraft should receive during the planning horizon.

(o, t): Dummy source and sink nodes of the network.

<u>Parameters</u>

DT_i: Local time when flight leg i depart from the airport, known as departure time.

O_{ia}: Binary indicator for the origin airport of flight leg i. It takes the value of 1 when the origin airport of flight leg i shares the same location as the airport a, and 0 otherwise.

AT_i: Local time when flight leg i arrives the airport, known as the arrival time.

D_{ia}: Binary indicator for the destination airport of flight leg i. It takes the value of 1 when the destination airport of flight leg i shares the same location as the airport a, and 0 otherwise.

FT_i: Duration of flight leg i.

TRT: Time consumed for getting passenger off, unloading the luggage, changing the gate, boarding, loading the luggage, and fueling the aircraft. This time is known as the turn -around time.

b_{ij}: Through value related to the connection between two consecutive flight legs i and j.

T_{max}: Maximum number of allowable flying time since last maintenance operation.

C_{max}: Maximum number of allowable take-offs since last maintenance operation.

MP_m: Available number of workforce teams in maintenance station m.

OT_m: The time when the maintenance station m opens

ET_m: The time when the maintenance station m closes.

Mb_{ma}: Binary indicator for maintenance station m. It takes the value of 1 when the location of the maintenance station m and the airport a are identical, and 0 otherwise.

MAT: Time taken to complete Type A maintenance check.

KT: Fleet size.

V: Maximum average number of maintenance operations that each aircraft in the fleet should receive. This value is determined by using the following equation; $V = \sum_{i \in NF} FT_i/(T_{max}\, KT)$

M: A big number.

PC_m: Penalty cost paid by airline to the maintenance station m for each aircraft exceeds the maintenance station workforce capacity.

Decision variables

$x_{ijkv} \in \{0,1\}$: Flight coverage decision variable. It takes value of 1 if aircraft k flies two successive flight legs i and j before receiving maintenance operation number v and 0 otherwise.

$y_{imkv} \in \{0,1\}$: Visiting maintenance station decision variable. It takes the value of 1 when flight legs i is flown by aircraft k then the aircraft proceeds to maintenance station m to receive maintenance operation number v and 0 otherwise.

$z_{mjkv} \in \{0,1\}$: Leaving maintenance station decision variable. It takes value of 1 when aircraft k leave maintenance station m to fly flight legs j after receiving maintenance operation number v and 0 otherwise.

$RTAM_{kv} > 0$: Time when the maintenance operation number v is completed for an aircraft k. After this time the aircraft can resume covering flight legs.

$ENOA_m > 0$: Number of aircraft that violates the maintenance station m's workforce capacity.

3.3.2 Modified connection network

The proposed model formulation is built by using the connection network, as it is one of the efficient networks used to represent the AMRP (Gopalan and Talluri, 1998, Haouari et al., 2012). The network consists of two main components: nodes and arcs. The nodes represent the flight legs, where the arcs denote the possible connection between the flight legs. One of the main features of this study is that all the operational maintenance restrictions are considered in the proposed OARP, in which the times and locations of the maintenance stations are determined. Towards this goal, the specific number of maintenance operations that each aircraft should receive is initially determined using three information: the total number of flying hours, the fleet size, and the maximum allowable flying time since last maintenance operation for each aircraft. Next, the specific number of maintenance operations for each aircraft is assigned to each aircraft in the fleet, such that all the operational maintenance restrictions are respected and monitored. In order to simultaneously assign flight legs for aircraft and schedule their maintenance operations, other arc and node types should be added to the connection network. For this purpose, we modify the original connection network by adding three types of arcs and two types of nodes, as shown in Figure 3.1. So, after modification, the node set includes flight leg node set (NF) and the maintenance station node set (MT), where the arc set includes the ordinary arc set (ORD), the maintenance arc set ($MAINT$) and the auxiliary arc set (AUX). The ordinary arc $ord(i,j) \in ORD$ can be used in three different situations; connecting flight legs i and j, connecting flight legs and source node while starting route construction for each aircraft in the fleet, and connecting flight legs and sink node while completing route construction for each aircraft in the fleet. The modified connection network includes the maintenance arc $maint(i,m) \in MAINT$ to help in connecting flight leg i and maintenance station m. Lastly, the auxiliary arc $aux(m,j) \in AUX$ is incorporated in the network, so that aircraft can resume covering flight leg j after leaving maintenance station m.

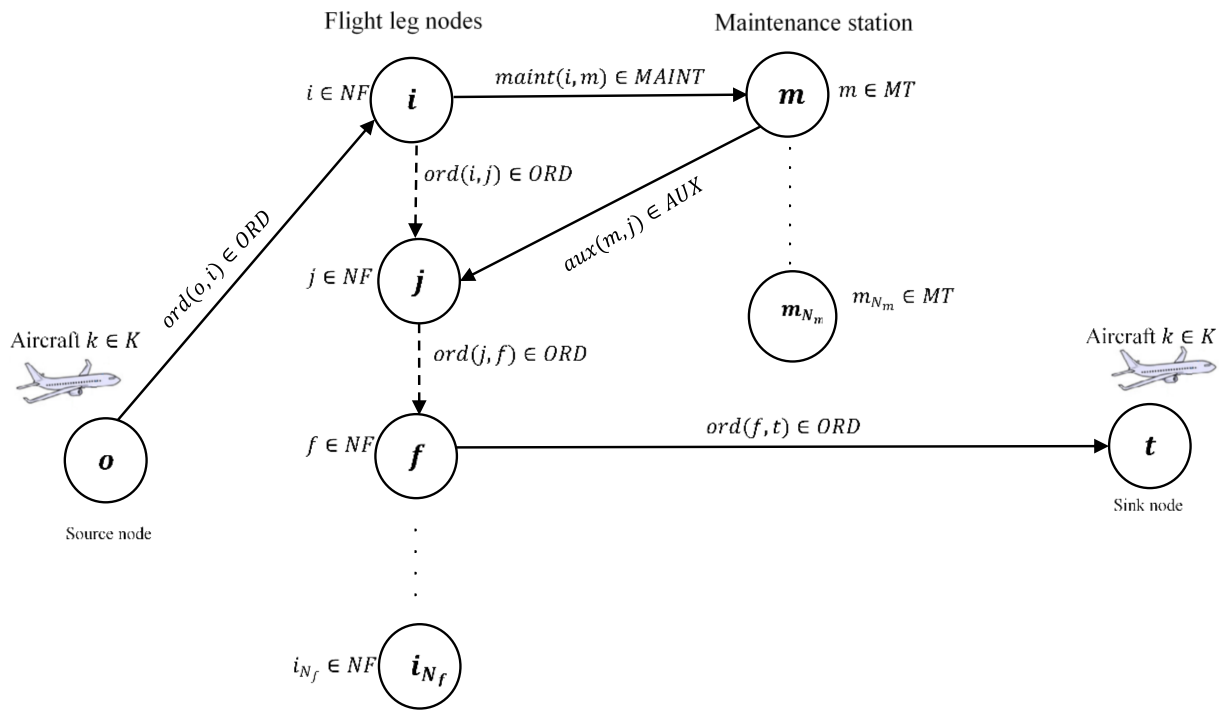


Figure 3.1: Representation of the modified connection network

3.3.3 Mathematical model formulation

The nodes and arcs included in the modified connection network ease formulating our proposed OARP model as a multi-commodity network flow-based MILP model, such that each aircraft represents a separate commodity moving throughout the network. To build the routing plan as an output for the OARP, five different decision variables are used. The first three decision variables are x_{ijkv}, y_{imkv}, z_{mjkv}, which represent the three main arc types explained in the previous section. The fourth decision variable is $RTAM_{kv}$. Indeed, this decision variable is incorporated in the model with the aim of calculating the suitable time for using the auxiliary arcs. As mentioned earlier, our model penalizes the airlines when the number of aircraft violates the workforce capacity of maintenance stations. For this purpose, the last decision variable $ENOA_m$ is cast in order to help in finding out the number of the aircraft that violates the workforce capacity of the maintenance station. By multiplying the $ENOA_m$ and the PC_m, the penalty cost can be easily calculated.

Based on the predefined notations, the mathematical model of OARP can be presented as follows:

Model 1

$$\max\ Z = \sum_{k\in K}\sum_{i\in NF}\sum_{j\in NF}\sum_{v\in V} b_{ij}\, x_{ijkv} - \sum_{m\in MT} ENOA_m PC_m \tag{3.1}$$

s.t.

$$\sum_{k\in k}\left(\sum_{j\in NF\cup\{t\}}\sum_{v\in V} x_{ijkv} + \sum_{m\in MT}\sum_{v\in V} y_{imkv}\right) = 1 \qquad \forall\ i \in NF \tag{3.2}$$

$$\sum_{j\in NF} x_{ojkv} + \sum_{m\in MT} y_{omkv} = 1 \qquad \forall\ k \in k, \forall\ v \in V \tag{3.3}$$

$$\sum_{i\in NF} x_{itkv} + \sum_{m\in MT} z_{mtkv} = 1 \qquad \forall\ k \in k, \forall\ v \in V \tag{3.4}$$

$$\sum_{j\in NF\cup\{o\}} x_{jikv} + \sum_{m\in MT} z_{mikv} = \sum_{j\in NF\cup\{t\}} x_{ijkv} + \sum_{m\in MT} y_{imkv}\ \forall\ i \in NF, \forall k \in k\,, \forall\ v \in V \tag{3.5}$$

$$\sum_{j\in NF}\sum_{v\in V} y_{jmkv} = \sum_{j\in NF\cup\{t\}}\sum_{v\in V} z_{mjkv} \qquad \forall m \in MT, \forall k \in k \tag{3.6}$$

$$AT_i + TRT - DT_j \leq M(1 - x_{ijkv}) \qquad \forall\ i \in NF, \forall j \in NF, \forall k \in k, \forall v \in V \tag{3.7}$$

$$\sum_{k\in k} x_{ijkv} \leq \sum_{a\in A} D_{ia} O_{ja} \qquad \forall\ i \in NF, \forall j \in NF, \forall v \in V \tag{3.8}$$

$$AT_i + MAT - ET_m \leq M(1 - y_{imkv}) \qquad \forall\ i \in NF, \forall m \in MT, \forall k \in k, \forall v \in V \tag{3.9}$$

$$OT_m - AT_i \leq M(1 - y_{imkv}) \qquad \forall\ i \in NF \cup \{o\}, \forall m \in MT, \forall k \in k, \forall v \in V \tag{3.10}$$

$$\sum_{k\in k} y_{imkv} \leq \sum_{a\in A} D_{ia} Mb_{ma} \qquad \forall\ i \in NF, \forall m \in MT, \forall v \in V \tag{3.11}$$

$$\sum_{k\in k} z_{mjkv} \le \sum_{a\in A} Mb_{ma}\, O_{ja} \qquad \forall m \in MT, \forall j \in NF, \forall v \in V \tag{3.12}$$

$$RTAM_{kv} \ge \sum_{i\in NF\cup\{o\}} \sum_{m\in MT} (AT_i + MAT)\, y_{imkv} \qquad \forall k \in k, \forall v \in V \tag{3.13}$$

$$RTAM_{kv} - DT_j \le M\left(1 - z_{mjkv}\right) \qquad \forall m \in MT, \forall j \in NF, \forall k \in k, \forall v \in V \tag{3.14}$$

$$\sum_{i\in NF\cup\{o\}} \sum_{j\in NF} x_{ijkv} \le C_{max} \qquad \forall k \in k, \forall v \in V \tag{3.15}$$

$$\sum_{i\in NF\cup\{o\}} \sum_{j\in NF} FT_j x_{ijkv} \le T_{max} \qquad \forall k \in k, \forall v = 1 \tag{3.16}$$

$$\sum_{i\in NF} \sum_{j\in NF} FT_j x_{ijkv} + \sum_{m\in MT} \sum_{j\in NF} FT_j z_{mjkv} \le T_{max} \qquad \forall k \in k, \forall v \in V/\{1\} \tag{3.17}$$

$$\sum_{i\in NF} \sum_{m\in MT} \sum_{v\in V} y_{imkv} \ge 1 \qquad \forall k \in k \tag{3.18}$$

$$\sum_{i\in NF\cup\{o\}} \sum_{k\in K} \sum_{v\in V} y_{imkv} \le MP_m \qquad \forall\, m \in MT \tag{3.19}$$

$$ENOA_m \ge \sum_{i\in NF\cup\{o\}} \sum_{k\in K} \sum_{v\in V} y_{imkv} - MP_m \qquad \forall\, m \in MT \tag{3.20}$$

$$x_{ijkv} \in \{0,1\} \qquad \forall\, i \in NF, \forall j \in NF, \forall k \in k, \forall v \in V \tag{3.21}$$

$$y_{imkv} \in \{0,1\} \qquad \forall\, i \in NF, \forall m \in MT, \forall k \in k, \forall v \in V \tag{3.22}$$

$$z_{mjkv} \in \{0,1\} \qquad \forall\, m \in MT, \forall\, j \in NF, \forall k \in k, \forall v \in V \tag{3.23}$$

$$RTAM_{kv} > 0 \qquad \forall k \in k, \forall \in V \tag{3.24}$$

$$ENOA_m \ge 0 \qquad \forall\, m \in MT \tag{3.25}$$

The objective function (3.1) maximizes the total profit as a difference between the revenue and the total penalty cost. One of the important issues that should be considered while constructing the routing plan is the coverage constraints, as casted in constraints (3.2) - (3.4). Constraints (3.2) ensure coverage each flight leg by exact one aircraft. Constraints

(3.3) indicate starting the route construction for each aircraft, meanwhile constraints (3.4) ensure completing the route construction for each aircraft.

In addition to the coverage constraints, it is important to keep the aircraft circulating throughout the network. For this purpose, our proposed model incorporates the balance constraints (3.5) and (3.6). Constraints (3.5) ensure the balance existence when covering flight legs nodes, i.e., when the aircraft uses the ordinary or the auxiliary arc to cover the flight leg, it is necessary for the aircraft to cover subsequent flight leg using the ordinary or the maintenance arc. Similar to constraints (3.5), constraints (3.6) ensure the balance when visiting the maintenance stations by the aircraft. Constraints (3.6) guarantee that when the aircraft uses the maintenance arc to covers the flight leg and proceeds to the maintenance stations, it is necessary for the aircraft to leave the maintenance station and cover subsequent flight leg using the auxiliary arc.

In order to use the ordinary arc to connect between two successive flight legs, these two flight legs should satisfy the time and place considerations, as indicated in constraints (3.7) and (3.8). Constraints (3.7) specify the time constraints such that two successive flight legs can be covered by the aircraft on the condition that the turn-around time is existed between the arrival and departure times of the first and second flight leg, respectively. The place constraints in (3.8) ensure that two successive flight legs can be flown by the same aircraft, on the condition that the same airport is shared by the destination and origin of the first and the second flight leg, respectively.

To use the maintenance arc while preparing the aircraft to visit the maintenance station, it is imperative to consider the place and time issues for two things: the potentially visited maintenance stations and the last flown flight leg. This is indicated in constraints (3.9) - (3.11). Constraints (3.9) and (3.10) indicate the time issue, as the maintenance stations working hours are considered. Constraints (3.9) ensure that maintenance station can be visited by the aircraft, on the condition that the closing time of the maintenance station is larger than or equal the arrival time of the last flown flight leg by the aircraft plus the Type A maintenance check duration. Same as constraints (3.9), Constraints (3.10) ensure that maintenance station can be visited by the aircraft, on the condition that the opening time of the maintenance station is less than or equal the arrival time of the last flown flight leg by the aircraft. Constraints (3.11) describe the place issues, and guarantee that the maintenance visit only occurs if the same location is shared by the location of maintenance station and the destination airport of the last flown flight leg.

After completion of the maintenance operation, the aircraft is required to leave the maintenance station and proceed flying the subsequent flight legs by adoption of the auxiliary arcs. To achieve this, constraints (3.12) - (3.14) are incorporated in the model, representing the time and place issues regarding this situation. The place issue is represented in constraints (3.12) that ensure the aircraft is only allowed to cover next flight legs after completing the maintenance operation on the condition that same location is shared by the maintenance station and the origin airport of the next flight leg. On the other hand, the time issue is represented in constraints in (3.14). These constraints ensure that the next flight leg can be proceed by the aircraft on the condition that $RTAM_{kv}$ of the aircraft is smaller than the departure time of the next flight leg. It should be noted that $RTAM_{kv}$ for each aircraft in the fleet is calculated in accordance to constraints (3.13).

To force the aircraft that need maintenance to receive such a maintenance operation, it cannot be achieved by adoption of the coverage and balance constraints. Therefore, we incorporate in the model the operational maintenance restrictions, as shown in constraints (3.15) - (3.18). Constraints (15) prohibit violation of the maximum number of take-offs restriction, where constraints (3.16) and (3.17) acts as the limiting constraints concerning the accumulated flying times since last maintenance operation. Constraints (3.18) are cast with the aim of ensuring a single maintenance visit for each aircraft in the fleet. Since our proposed model adopts a planning horizon of 4-day, and constraints (3.18) ensure at least a single maintenance visit for each aircraft, so the model satisfies the operational maintenance restriction regarding a maintenance visit every four days.

Checking the existence of sufficient workforce capacity in maintenance stations is very important while scheduling maintenance visits for the aircraft. Towards this goal, the workforce capacity constraints are formulated in the model, as shown in constraints (3.19). Using these constraints help in avoiding the overcapacity problem, as they ensure that maintenance station workforce capacity is not violated by the number of aircraft that visit the maintenance station. To calculate the number of aircraft that violates the maintenance station workforce capacity, we formulate constraints (3.20). Finally, constraints (3.21) - (3.25) represent the domain restrictions imposed upon the decision variables.

After presenting our proposed model, it is important to highlight the main differences between our model 1, and the others reported in the literature. For this purpose, we summarize these differences as follows:

- The proposed model 1 considers the maintenance station workforce capacity, as shown by Eq. (3.19). Indeed, this consideration was neglected in the literature, as in the model presented by Barnhart et al. (1998). In the study by Haouari et al. (2012), the maintenance station workforce capacity was considered in their proposed model, but was relaxed in the computational experiments part. In addition, the model 1 pays attention to the penalty cost paid by the airlines to the maintenance providers in case of assignment aircraft to maintenance stations that suffer from an insufficient workforce capacity. Overlooking the workforce capacity would result in assignment aircraft to maintenance stations that own insufficient workforce capacity, which in turn results in increasing the waiting time for the aircraft before receiving the maintenance operation. In order to avoid this situation, the maintenance stations deploy more hands to maintain the excess aircraft, leading finally to an additional penalty cost. This description reveals the importance of considering the restriction of the workforce capacity of maintenance stations, which helps in avoiding the prescribed situation, resulting in a significant decrease in the penalty cost.
- The proposed model 1 considers the maintenance stations working hours, as shown by Eqs. (3.9) and (3.10). To our best knowledge, this consideration has not been considered in the previous studies (Sriram and Haghani, 2003, Sarac et al., 2006, Haouari et al., 2012, Başdere and Bilge, 2014). Overlooking this consideration

might cause arriving the aircraft on different times than the maintenance stations working hours, resulting in a long waiting time for the aircraft in receiving maintenance service. Consequently, the subsequent scheduled flights are cancelled, resulting in an additional cost to recover the cancelled flights. Accordingly, taking into account the maintenance stations working hours helps airlines to avoid flight cancelations, which in turns causes a decrease in the operational costs.

- In the literature, to simplify the OARP formulation, it was assumed that overnight is the only time for carrying out the maintenance operations (Liang et al., 2011, Sriram and Haghani, 2003). In real practice, the situation is different as the maintenance operations can be carried out during the maintenance stations working hours. Usually, the maintenance stations working hours cover 24 hours of the day, including two types of shifts; daytime shift and the overnight shift. Following the assumption of carrying out the maintenance operation only at overnight shift, while overlooking the daytime shift, meaning that any aircraft that arrive at the morning have to wait at the airport till night before receiving the maintenance operation. This results in a long waiting time for these aircraft, leading to a cancelation for subsequent flights to be covered by the aircraft. Towards the goal of avoiding this situation, the proposed model 1 considers the working hours of the maintenance stations, which means adding the daytime shift besides the overnight shift. By doing so, the long waiting time for the aircraft that arrive in the morning can be avoided as these aircraft can receive the maintenance service during the daytime shift, leading finally to a reduction in the number of cancelled flights.

3.3.4 Complexity analysis

It is commonly known that the complexity of the mixed integer programming models can be determined according to the number of used decision variables and constraints in the proposed model (Dong et al., 2016, Liang et al., 2011). Following this way of figuring out the complexity, it is revealed that the number of decision variables required for the proposed model 1 is $|K| \times |V| \times (|NF|^2 + 2|NF| \times |MT| + 1) + |MT|$. On the other hand, the number of constraints in the proposed model 1 is at most $O(|NF|^2 \times |K| \times |MT| \times |V|)$. To clarify this point in terms of figures, a simple example is provided. Suppose that we have 8 aircraft that should cover 40 flight legs, and these aircraft can visit 4 maintenance stations to receive the maintenance operations, which are fixed to be two operations during the planning horizon. To use the proposed model 1 in handling this test instance, we need $8*2*(40^2 + 2*40*4 + 1) + 4 = 30740$ decision variables, whereas the maximum number of required constraints is $O(40^2 *8*4*2)$. It should be noted that the previous example is the smallest test case adopted in this study.

The above-mentioned description indicates that the $O(|K| \times |V| \times |NF|^2)$ can be used to express the state complexity of the model. This means that $|K|$ aircraft cover $|NF|$ flight legs exactly once, meanwhile each aircraft visits maintenance station $|V|$ times. By looking at the problem description in our hand, we can discover that this description matches the description of the partition problem (Başdere and Bilge, 2014, Sarac et al., 2006). Since the proposed model 1 contains a partition problem that is known as NP-complete, our model is NP-hard.

In an attempt to demonstrate our model formulation scale advantage, a comparison between the complexity of set-partitioning partitioning formulation $O(2^{|NF|})$ and the space complexity of our model $O(|K| \times |V| \times |NF|^2)$ is conducted. This comparison reveals better scalability of our polynomial model formulation over the set-partitioning formulation. The rationale behind this better scalability lies in that the set-portioning formulation generates all possible feasible routes, which are significantly more than the decision variables used in our model.

From the comparison, it is clear that our polynomial formulation is more scalable than the set-partitioning formulation. However, it does not mean that a commercial optimization software like CPLEX can be directly used to solve our model in a reasonable computational time. In our preliminary results, CPLEX showed a good performance in solving small size test instances. However, it is challenging for CPLEX to solve medium and large-scale test instances, as the feasible solutions are difficult to achieve. Therefore, to solve medium and large-scale test instances, an efficient algorithm is developed in the following section.

3.4 Solution Method

This section presents our solution method, but before doing so, we discuss the solution methods used to solve the existing OARP. To solve OARP, there are two main methodologies. The first methodology is to develop a set-partitioning formulation for OARP, then adopt the column generation in order to solve the proposed formulation, as shown in the study by Sarac et al. (2006). Using column generation as a solution method for our proposed model necessitates developing the model based on set-partitioning formulation. To do so, we need to generate all the feasible routes. As we mentioned earlier, the set-partitioning generates all the feasible routes, which grows exponentially with the number of flight legs. Therefore, solving medium and large-size problems in a reasonable computational time becomes a great challenge. Since our target is to solve real and large-size test cases, therefore, it is not suitable to use column generation to solve for our proposed model. The finding is reported in the work by Sarac et al. (2006), in which column generation was adopted to solve the proposed model. This soution method showed a good performance for solving a case of 175 flight legs, but it could not solve larger cases. The second methodology is to develop a multi-commodity network flow formulation for the OARP, and adopt different tools as shown in different studies, such as Sriram and Haghani (2003), Haouari et al. (2012) and Başdere and Bilge (2014). In the study by Sriram and Haghani (2003), two models were developed. To solve the first model, an effective algorithm based on depth search and random search was developed. This effective algorithm solved only cases with size up to 58 flight legs but failed to solve any cases with larger size. Based on this performance, it is not useful to apply that algorithm as a solution method for our proposed model, as our target is handling cases with size up to 400 flight legs as a real case, and up to 4000 flight legs as a generated case. The study by Sriram and Haghani (2003) presented a second model for OARP, in which the cumulative flying hours was considered. However, they did not attempt to solve it. It is worth mentioning that our proposed model is more complicated compared to the second model proposed by Sriram and Haghani (2003). This complication stems from the fact that all the operational maintenance restrictions mandated by FAA, besides the maintenance station workforce capacity and working hours, are considered in our model, whereas the second model proposed by Sriram and Haghani (2003) overlooks some restrictions such as the maximum number of take-offs and one maintenance visit every four days, which in turn results in a reduction in the number of constraints and decision variables. For the work by Haouari et al. (2012), CPLEX 12.1 was adopted to solve their model. Indeed, CPLEX is also adopted to solve our proposed model, while handling small size test instances. Lastly, in the study by Başdere and Bilge (2014), two different solution methods were used: branch and bound (B&B) for small scale test instances and compressed annealing (CA) for large scale test instances. Applying B&B as a solution method for our proposed model is not promising as B&B takes long computational time and sometimes feasible solutions are difficult to be achieved for medium and large-scale problems. For CA, on the other hand, it can be adopted to solve simple models, in which the number of decision variable and constraints are not large. Therefore, for our model that contains relatively large number of decision variables and constraints, it becomes a challenge for CA to solve this model.

The previous studies reveal that it is relatively easier to solve the existing OARP as these models were developed based on the multi-commodity network flow formulation with relatively low number of decision variables and constraints. So, it is an easy mission to solve the existing models. Although solving the existing model is an easy mission, handling our model is much trickier because of two issues. Firstly, all the operational maintenance restrictions mandated by FAA, besides the maintenance stations workforce capacity and working hours, are considered in our model. Due to all these considerations, more decision variables and constraints are added, so that it is expected for computational time to be long as our model is shown to be a NP-hard. Secondly, in the literature, the original structure of multi-commodity network flow formulation can be represented by only x_{ijkv} as a decision variable and its related constraints, as described by Eq. (3.2)- (3.5), (3.7) and (3.8). Considering all the operational maintenance restrictions results in incorporation of new decision variables (y_{imkv}, z_{mjkv}, $RTAM_{kv}$, and $ENOA_m$) in our proposed model. These new decision variables in turn result in new terms being added to Eq. (3.2)- (3.5), and new constraints are cast, as described by Eq. (3.6) and (3.9)- (3.20). Indeed, the original structure of multi-commodity network flow formulation is seriously destroyed because of adding the new decision variables and their related constraints, resulting in a difficult task to solve the proposed model. The previous two finding are confirmed in the work by Sriram and Haghani (2003), in which the authors tried to adopt the genetic algorithms to solve their proposed model.

Based on the above discussion, an efficient solution algorithm is proposed in order to solve model 1. It is noticed after constructing our model that it is a challenge to construct aircraft routes while simultaneously focusing on two issues of maximizing profit and respecting all the operational maintenance restrictions. Consequently, the proposed algorithm divides this mission into two main procedures. Firstly, constructing sub-routes with maximized profit, while taking into consideration the coverage, balance, time and place constraints as described by Eqs. (3.2) -(3.8). Then, secondly, using the constructed sub-routes to construct complete routes that satisfy all the operational maintenance restrictions stated in Eqs. (3.9) -(3.20). The detailed procedures of the algorithm can be summarized in the following steps:

Step 0: Design two lists, such that the first list contains the aircraft (K), whereas the second list contains the flight leg nodes (NF).

Step 1: By adopting the following rule; $V = \sum_{i \in NF} FT_i / (T_{max} KT)$, calculate the maximum average number of maintenance operations (V) that each aircraft should receive during the planning horizon.

Step 2: Divide the NF list into two; the star list (SL) and the normal list (NL). For SL list, it contains the through connects, which allow us to give this list high priority when constructing the aircraft routes. On the other hand, NL list contains the remaining flight legs; thus, low priority is given to this list when constructing the aircraft routes. Towards the goal of dividing NF into two lists, we calculate the connection time between each pair of flight legs stored in the NF list. If the pair is a through connect as its connection time has a through value, then this pair should be removed form NF and should be stored in SL. For the rest of flight legs, store them in NL.

Step 3: Construct the sub-routes list (SRL) by using the pairs stored in the SL list. To do so, it is necessary to connect two pairs from SL, such that the ending flight of the first pair is the same as the starting flight of the second pair. After constructing the sub-routes, store it in SRL. Of course, it is difficult to connect all the pairs existed SL, therefore the non-connected pairs should be stored in SRL. So, we have three lists, K, SRL, and NL, by the end of this step,

Step 4: Set the number of iterations to be one.

Step 5: Check the status of the K list. In the case of nonempty list, select kth aircraft and proceed to step 6, otherwise move to step 8.

Step 6: Initiate the process of constructing complete route for the kth aircraft by following backward and forward insertion approaches, such that the operational maintenance restrictions described in Eqs. (3.9) -(3.20) are taken into consideration. The backward insertion approach can be carried out by following the next sub-steps:

Step a: Check the condition of SRL, as it has high priority. In the case of nonempty list, select one element randomly, otherwise select that element from the low priority NL. The selected element constitutes the first part of the constructed complete route.

Step b: Determine the starting flight leg for the selected element. To do so, we need to determine whether the element is selected from SRL or NL. If the element is selected from SRL, then the starting flight leg is the first flight leg of the selected element. On the other hand, if the element is selected from NL, then the starting flight leg is the element itself.

Step c: Scan through SRL to find suitable to be inserted backwardly to (before) the selected element. If SRL is empty, we shift our scan to be done through NL. During the scan process, the place and time restrictions stated in Eqs. (3.7) and (3.8) should be taken into consideration. In the case of empty SRL and NL, proceed to step i.

Step d: Identify the possible elements that can be inserted backwardly to the selected element. For each possible element, determine its connection time and its related through value. If there are no possible elements for backward insertion, then proceed to step h, otherwise proceed to step e.

Step e: Among the possible elements for backwards insertion, select the one that has the highest through value. Then, determine if there is a violation of the operational maintenance restriction described by Eqs. (3.15) -(3.18) after selecting the possible element. Proceed to step f in the case that these constrains are violated, otherwise move to step g.

Step f: Use as a basis the maintenance stations working hours constraints shown in Eqs. (3.9) and (3.10), the location of maintenance stations constraints shown in Eq. (3.11), and the maintenance station workforce capacity constraints stated in Eq. (3.19), in order to draw up a maintenance visit. On the completion of the maintenance operation, use the constrains described by Eqs. (3.4) -(3.6) and Eqs. (3.12) -(3.14) to select suitable element from SRL or NL. Then, go to step b.

Step g: Remove the selected element from SRL or NL, and add that selected element to the constructed route.

Step h: Identify the new staring flight leg, then proceed to step c.

Step i: Terminate the backward insertion approach.

All the previous sub-steps indicate the procedure followed while adopting the backward insertion. Next, it is the turn of the forward insertion approach to be conducted. To do so, we follow the same sub-steps carried out in the backward insertion approach, except two different changes. Firstly, rather determining the starting flight leg for the selected element as in the backward insertion, in the forward insertion, the ending flight leg should be determined for the selected element in step b. If the element is selected from SRL, then the ending flight leg is the last flight leg of the selected element. On the other hand, if the element is selected from NL, then the ending flight leg is the element itself. Secondly, instead of inserting the elements backwardly, the elements are inserted forwardly to (after) the element that is selected in step a.

Step 7: Terminate the route constructed for kth aircraft, then proceed to step 5, after excluding the aircraft k from the K list.

Step 8: For the current iteration, calculate the solution, then update the best solution.

Step 9: Check the satisfaction status of the stopping criteria. In the case of satisfied stopping criteria, terminate the algorithm, otherwise proceed to step 5 after increasing the number of iterations, and updating the empty lists of K, SRL, and NL by using the same lists stored in step 3.

Figure 3.2 elaborates the algorithm procedure flow chart, which is terminated when the stopping criteria is satisfied. In this algorithm, the stopping criteria can be satisfied in three different situations; (1) For small test cases, when the current solution reaches the exact solution. (2) for medium and large test cases, when the current solution reaches the best upper bound. (3) when the maximum number of iterations is exceeded by the current number of iterations. It should be noted that the maximum number of iterations is set at 1000 in all test instances.

After adopting our proposed algorithm, one of the questions that might be asked is "how the performance of the proposed algorithm can be evaluated". To answer this question, we propose making a comparison between the solutions obtained from our proposed algorithm and the exact solution obtained from CPLEX, while solving small test cases. For medium and large test cases, on the other hand, we propose using the best upper bound (UB) generated by CPLEX as a criterion to evaluate the proposed algorithm performance. This is because even the feasible solutions are difficult to be obtained by CPLEX. In this study, the UB is obtained by setting the maximum CPU time for CPLEX to be 6 hours. Indeed, this period is selected because longer time does not produce a bound with better quality.

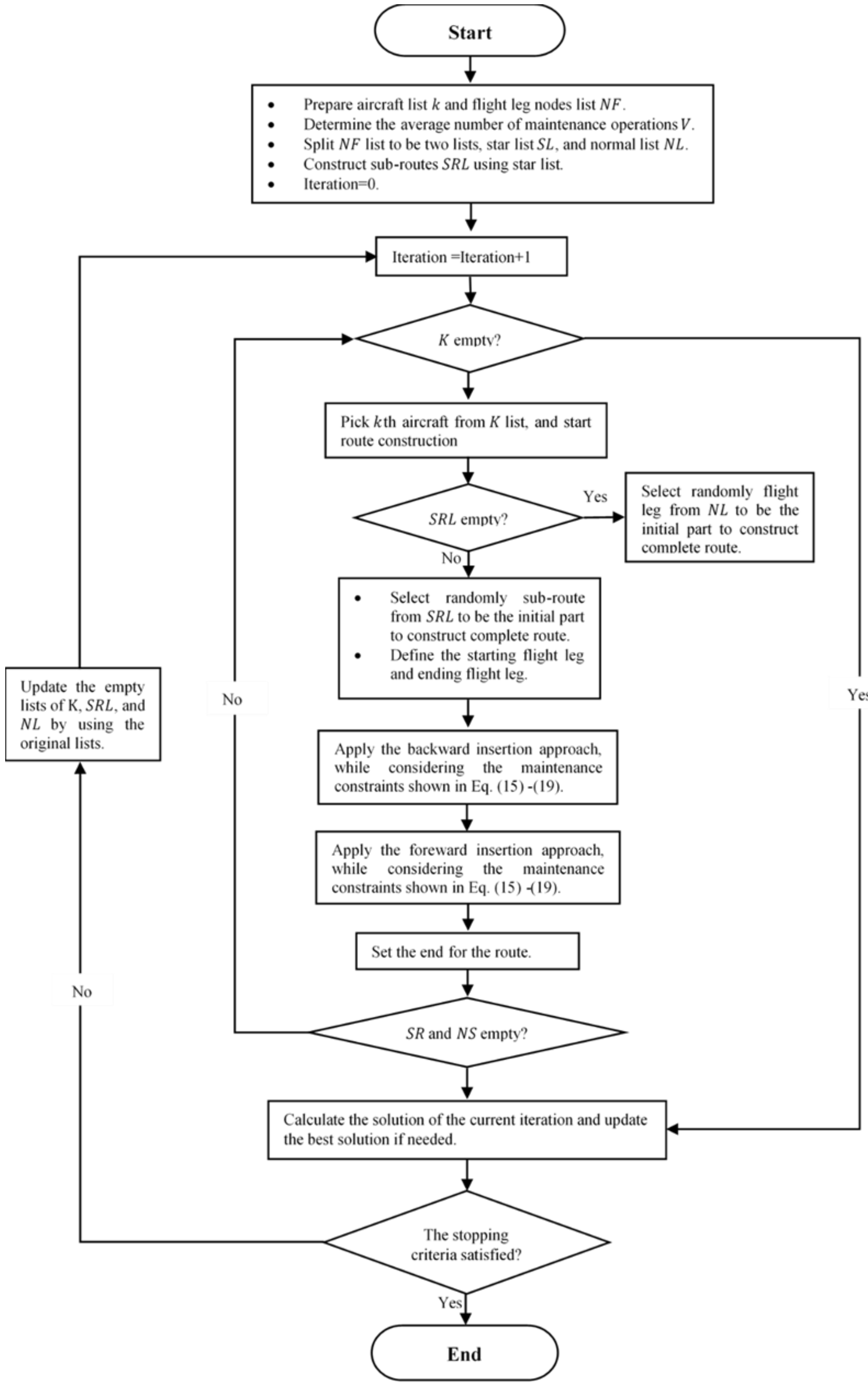


Figure 3.2: Flowchart of the solution algorithm

3.5 The Proposed Model 1 and Existing Models

In this section, we make a comparison between model 1and the existing models in the literature with the aim of testing the implications of considering the maintenance stations workforce capacity on the profitability. To do so, model 1 was modified in order to be approximately like the models presented in the work by Haouari et al. (2012) and Başdere and Bilge (2014). The modified model is called model 2, in which the objective function and constraints of model 1 are considered, except ignoring the maintenance stations working hours constraints and the maintenance stations workforce capacity constraints, as stated in Eqs. (3.9), (3.10), (3.13), and (3.19).

Since the algorithm presented in the previous section efficiently solves model, we applied the same algorithm to solve model 2, but after making a small modification. This modification includes ignoring the maintenance stations working hours constraints and the maintenance stations workforce capacity constraints, throughout the whole algorithm.

3.6 Computational Results

After developing the effective solution algorithm, it is necessary to demonstrate the effectiveness of this algorithm. Therefore, we conduct computational experiments based on real data obtained from a Middle Eastern major airline. In this section, we report the results obtained from the experiments, while using our proposed algorithm and CPLEX 12.1.

3.6.1 Test cases

The experiments of this study were conducted based on fifteen test instances. These test instances are divided into real and generated cases, such that the first ten cases are real cases obtained from a Middle Eastern major airline, whereas the rest five cases are generated based on the combinations of the first ten cases. For real cases, they are constructed by using ten real flight schedules flown by different fleets. To generate larger test cases for testing purposes, SIM01, SIM02, SIM03, SIM04, and SIM05 are constructed using different ways. For example, SIM01 and SIM02 are built by merging the flight schedules of multiple fleets, such that SIM01 is built by merging the flight schedules of cases 7 and 9, whereas SIM02 is built by merging the flight schedules of cases 9 and 10. Another way for constructing large test cases, is doubling the flight schedule of some cases by adjusting the arrival and departure times of the flight schedule using a specific period. In such a way, SIM03 is constructed by doubling the flight schedule of SIM02 twice through adjusting the arrival and departure times of the flight schedule by 15 minutes. Similarly, SIM04 is generated by doubling the flight schedule of SIM01 four times through adjusting the arrival and departure times of the flight schedule by 15, 30, 45, and 60 minutes, for each time of duplication. Finally, SIM05 is built by doubling the flight schedule of SIM04 twice through adjusting the arrival and departure times of the flight schedule by 5 hours. It is interesting to mention that SIM05 is even larger than the size of the largest fleet in the world, named Southwest Airline Boeing 737-700, which include 350 aircraft to cover 3469 flight legs in 4 days (Liang et al., 2015). Table 3.1 presents the complete feature for the test instances.

For the whole experiments, it was recommended by the airline that the turn-around time TRT should take the value of 45 minutes, T_{max} should take the value of 40 hours, and the time taken to complete Type A maintenance check should be within 8 hours. On the other hand, regarding to the through values, it was assumed by the airline that these values appear when the transit time or connection time ranges from 45 minutes to 1.5 hour. Finally, we set the penalty cost to be around 500.

It is commonly known that the runs of any solution method should be replicated several times to evaluate the performance of the algorithm. Therefore, we replicate the runs of the proposed algorithm thirty times for all test cases. We decide thirty runs as additional runs do not demonstrate better results. Note that these experiments were performed on 8 GB RAM Windows 10 laptop, equipped with an Intel i7 CPU with a clock speed of 2.50 GHz. MATLAB R2014a was used to code the proposed model and the effective algorithm.

Table 3.1: Features of test cases

Test cases	Number of flight legs	Fleet size	Maximum number of take-offs	Number of airports	Maintenance Stations
Case 1	40	8	10	4	4
Case 2	48	7	7	5	4
Case 3	64	8	7	7	4
Case 4	96	14	10	13	6
Case 5	120	13	10	8	6
Case 6	160	11	15	10	6
Case 7	200	15	15	8	9
Case 8	240	26	15	19	9
Case 9	296	30	15	26	9
Case 10	400	42	15	28	18
SIM01	496	45	15	33	18
SIM02	696	72	15	53	27
SIM03	1392	144	15	53	27
SIM04	1984	180	15	33	30
SIM05	3968	360	15	33	30

3.6.2 Results of small size test instances

Table 3.2 shows the results obtained from the proposed algorithm replications and CPLEX, while solving small size test instances, expressed through the first eight cases of Table 2. The results of CPLEX are presented in the first two columns of the Table 3.1, including the optimal solution (Z*) and the computational time CPU(s). On the other hand, the results of the proposed algorithm are summarized in the remaining columns of Table 3.2. The best solution of the algorithm replication is represented by the Z_{best} column, whereas the average and the standard deviation summaries of the replication results are reported in the $\bar{Z}$ column and $\sigma_z$column, respectively. Lastly, the average computational time of the proposed algorithm is recorded in the $\overline{CPU(s)}$ column. To obtain the computational time of the proposed algorithm, the internal calculation function of MATLAB is used. Towards the goal of evaluating the proposed algorithm performance, we select the relative difference between the optimal solution obtained by CPLEX and the average solution obtained by the proposed algorithm, to be the performance indicator. This performance indicator is known in the literature as the optimality gap (%Difference), which can be determined as $(Z^* - \bar{Z})/ Z^*$.

Table 3.2: Results of small size cases

Test cases	**CPLEX**		**Proposed algorithm**				**%Difference**
	Z^*	$CPU(s)$	Z_{best}	$\bar{Z}$	σ_z	$\overline{CPU(s)}$	
Case 1	16,667	1.44	16,667	16,667	0	0.28	0
Case 2	2,333	3.08	2,333	2,333	0	0.30	0
Case 3	5,333	18.19	5,333	5,333	0	0.25	0
Case 4	10,000	53.06	10,000	10,000	0	0.26	0
Case 5	15,000	243.80	15,000	14,909	62.27	0.84	0.61
Case 6	22,000	372.05	22,000	21,852	95.92	1.56	0.67
Case 7	42,667	633.88	42,667	42,542	139.70	1.61	0.29
Case 8	34,083	9130.19	34,083	33,899	183.45	2.64	0.54

By looking at the results obtained by the proposed algorithm, we can see its Z_{best} and $\bar{Z}$ reach the optimal solution Z^*, especially in case 1 up to case 4. With respect to the remaining cases, the performance of the proposed algorithm in terms of Z_{best} still reach the optimal solution Z^*, but its average solution $\bar{Z}$ deviates mostly with 0.67% from the optimal solution Z^*, as shown in case 5 up to case 8. The standard deviation reported in Table 3.3 reveals no solution variability for case 1 up to case 4, but for the reaming cases, this variability slightly increases. This performance indicates the reliability and the stability of the proposed algorithm.

By looking precisely at the computational time for both solution methods reported in Table 3.3, we can notice the fast performance of the proposed algorithm compared with CPLEX. This appears significantly when the proposed algorithm can find the solution for case 8 within almost 3 seconds, whereas 2.5 hours is spent by CPLEX to get the solution for the same case. For all eight cases discussed in this section, the computational time comparison taken to solve these cases using the two solution methods is shown in Figure 3.3.

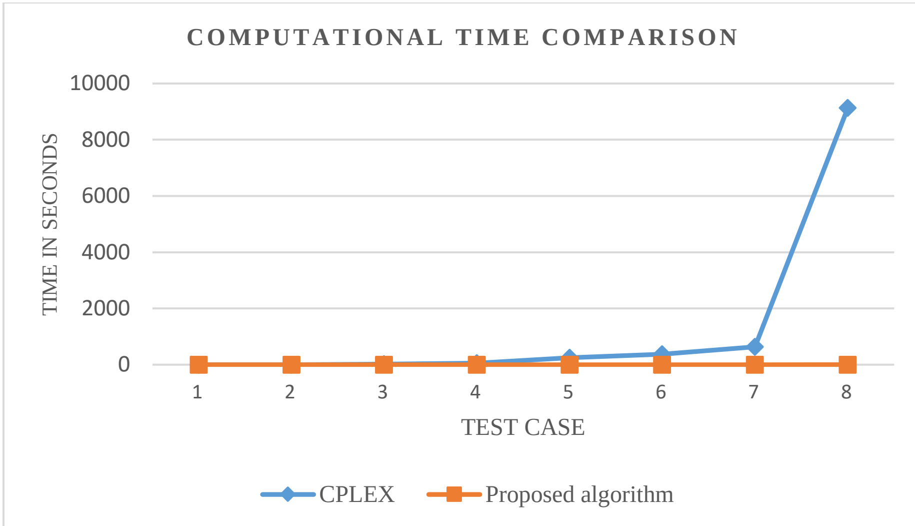


Figure 3.3: Computational time for CPLEX and proposed algorithm

In this section, the discussed results reveal that, in terms of computational time, the proposed algorithm outperforms CPLEX in handling small sized test instances. This underscores the significant time saving achieved when using the proposed algorithm. Meanwhile, the proposed outperforms CPLEX in terms of solution quality, as the best solution always equals the optimal solution and the average solution slightly deviates by almost 0.67%. from the optimal solution

In this section, the proposed algorithm shows a good performance when solving small test cases with sizes up to 240 flights and 26 aircraft. Indeed, using these test instances enables us to benchmark the results of the proposed algorithm with the optimal solutions. However, solving these test instances is not large scale enough to demonstrate the potential and applicability of the proposed algorithm. In this connection, we decide to use medium and large size test instances, which cannot be solved by CPLEX as feasible solution is even difficult to be obtained within a reasonable computational time. This result in a difficulty of computing the optimality gap as optimal solution cannot be measured. Therefore, we decide to use the (%GAP) as a performance indicator. Note that (%GAP) can be defined as the relative difference between CPLEX's result in terms of the best upper bound (UB) and the proposed algorithm's results in terms of average solution, as a performance indicator. Another advantage from using medium and large size test instances is testing the potential of the proposed algorithm to handle real life problems. In the light of the previous observations, we extend our computational experiments, in which the proposed algorithm can be tested while solving medium and large size test instances.

3.6.3 Results of medium and large size test instances

The experiments discussed in this section are conducted based on the cases 9, 10, SIM01, SIM02, SIM03, and SIM04 presented in Table 3.1. Using such test instances, as being medium and large size test instances, help us in carrying out experiments with the objective of evaluating the scalability of the proposed algorithm to solve real live problems. The results of these experiments are illustrated in Table 3.3, which reports the same statistics presented in Table 3.2. It should be noted that in this section, the UB was obtained by running CPLEX for 6 hours.

By looking at Table 3.3, it is obvious that the performance of the proposed algorithm is promising. This is because the proposed algorithm can find out high quality solutions in a relatively short computational time. For the computational time, the proposed algorithm shows fast performance, as it solves the largest case, which is SIM04, within 4 minutes. Meanwhile, with respect to the solution quality, Z_{best} of the proposed algorithm reaches UB in all cases, whereas $\bar{Z}$ deviates with a %GAP of almost less than 0.7 %. from UB. These experiments reveal the potential of the proposed algorithm to handle large size and real-life problems, since it can generate profitable routes in a short computational time. If we look at the standard deviation in Table 3.3, we can notice low solution variability. This again confirms that the proposed algorithm is stable and reliable even when solving large size test instances.

Table 3.3: Results of medium and large size cases

Test cases	UB	Proposed algorithm				%GAP
		Z_{best}	$\bar{Z}$	σ_z	$\overline{CPU(s)}$	
Case 9	60,333	60,333	59,997	325.77	2.57	0.55
Case 10	72,583	72,583	72,097	448.00	9.22	0.66
SIM01	111,000	111,000	110,425	385.80	5.22	0.52
SIM02	140,916	140,916	140,117	631.23	20.07	0.56
SIM03	297,833	297,833	296,175	1,500.23	78.18	0.55
SIM04	486,250	486,250	483,533	3,176.70	224.00	0.55

3.6.4 Results of SIM05 test instance

In this section, we extend our experiments to test the potential of the proposed algorithm to solve the SIM05 test instance, which is larger than the size of the largest fleet in the world. Although the test instance SIM05 is very large, the results of the proposed algorithm are promising, as shown in Table 3.4. The proposed algorithm successfully solves this case with high quality solutions as the Z_{best} reaches the UB, whereas $\bar{Z}$ deviates with a %GAP of 0.83%. from UB. Regarding the computational time, the fast performance of the algorithm enables solving the case within 35 minutes. These results reveal the potential of the proposed algorithm to handle very large test instances as it provides high quality solutions in a short computational time.

Table 3.4: Results of SIM05 test instance

Test case: SIM05	
UB	1,052,250
Z_{best}	1,052,250
$\bar{Z}$	1,043,516
σ_z	8,526.38
$\overline{CPU(s)}$	2100.00
%GAP	0.83

3.6.5 Performance analysis

So far, the proposed algorithm performance is discussed when the real-life test instances are solved. However, this is not enough to show the advantage of the algorithm over the solution methods existed in literature. In this connection, in this section, our experiments were extended in with the objective of making a comparison between the performance of our proposed algorithm and another two solution methods. The first solution method was adopted in the work by Haouari et al. (2012) and is called CPLEX, whereas the second solution method was proposed in the work by Başdere and Bilge (2014) and is called compressed annealing (CA). We select CA for this comparison due to its good performance in handling large scale test instances, as stated by Başdere and Bilge (2014). Based on this observation, CA is worth enough to be selected for our comparison.

As mentioned earlier, the proposed model 1 includes more feature than those included in the existing models. So, selecting model 1 to be the model used in comparing the performance of the aforementioned three solution methods, will favour our algorithm over the rest of other solution methods. This is because we tailored our proposed algorithm to solve model 1. In this connection, in order to conduct a fair comparison, model 2 was selected to be solved by the three solution methods, as model 2 includes the similar characteristics as those included in the existing models. To solve model 2 using our algorithm, we follow the modifications explained in section 3.5. Meanwhile, to solve model 2 using CA, the same procedure and parameter setting developed by Başdere and Bilge (2014) were followed. By using all cases presented in Table 3.1, our experiments are performed, and their results are summarized in Table 3.5, which reports statistics that are similar to those reported in previous sections.

Table 3.5: Results of CPLEX, CA, and proposed algorithm when solving model 2

Test Cases	CPLEX		Compressed Annealing (CA)				The proposed Algorithm				
	Z^*	$CPU(s)$	Z_{best}	$\bar{Z}$	σ_z	$\overline{CPU(s)}$	Z_{best}	$\bar{Z}$	σ_z	$\overline{CPU(s)}$	$IMP_{CA}(\%)$
Case 1	16,157	0.96	16,157	16,157	0	0.87	16,157	16,157	0	0.22	0
Case 2	2,258	2.85	2,258	2,258	0	1.93	2,258	2,258	0	0.28	0
Case 3	5,151	16.23	5,151	5,151	0	12.28	5,151	5,151	0	0.23	0
Case 4	9,655	50.08	9,655	9,655	0	31.49	9,655	9,655	0	0.23	0
Case 5	14,533	220.95	14,225	14,150	411.79	70.23	14,533	14,348	54.30	0.79	1.37
Case 6	21,337	333.07	21,141	20,796	224.48	220.47	21,337	20,950	92.87	1.44	0.73
Case 7	40,995	590.78	39,315	39,220	283.64	350.48	40,995	40,751	123.58	1.57	3.75
Case 8	32,508	8598.32	31,283	31,099	430.69	780.09	32,508	32,441	160.28	2.50	4.13
Case 9	--	--	56,210	55,320	582.34	1528.23	57,452	57,321	289.74	2.33	3.49
Case 10	--	--	67,021	65,789	865.12	1768.37	69,158	68,853	440.52	8.78	4.45
SIM01	--	--	102,987	100,254	878.28	2100.57	105,750	105,401	320.78	4.98	4.88
SIM02	--	--	127,897	122,587	1,700.64	3150.91	132,008	131,486	532.03	18.20	6.76
SIM03	--	--	257,892	250,478	2,271.33	5780.41	271,589	270,230	1,328.70	67.52	7.30
SIM04	--	--	402,874	388,754	4,504.66	7381.78	420,986	418,836	3,278.24	198.03	7.18
SIM05	--	--	783,254	750,147	10,254.12	18320.38	860,789	847,544	6,324.51	1824.00	11.4

Note: $IMP_{CA}(\%) = (\bar{Z}_{proposed\ algorithm} - \bar{Z}_{CA})/\bar{Z}_{proposed\ algorithm}$

By looking at the Table 3.5, we can see that the results reported for the proposed algorithm and CPLEX are almost the same as those discussed in section 3.6.2. On the other hand, by comparing the results obtained from the proposed algorithm and CA, we can notice a clear outperformance of the proposed algorithm over CA in two different aspects: solution quality and computational time. With respect to the solution quality, the performance of both methods is the same as both provide the same $\bar{Z}$, while handling small size instances as in cases 1 up to case 4. For large test instances as in the remaining of the cases, the performance of the proposed algorithm is better than CA. This is appeared clearly because $\bar{Z}$ of CA is lower than $\bar{Z}$ of the proposed algorithm. The outperformance ratio (IMP_{CA}) starts to be about 1.37% as shown by case 5 and rises gradually up to be around 11.4% as shown by the last case, called SIM05. For computational time, the first five cases do not show a significant difference in the performance of both of CA and the proposed algorithm. But, the proposed algorithm is much faster than CA, when handling the rest of the cases. This fast performance is clearly demonstrated in case SIM05, since the solution is produced by the proposed algorithm within 30.4 minutes while CA needs up to 5 hours.

One of the questions that might be asked is "what the reasons behind the outperformance of the proposed algorithm over CA are". The answer to this question can be briefed as follows:

- *Solution quality:* the first step of CA is building an initial solution. To do so, CA uses a simplified version of OARP. In particular, this simplified model ignores the objective function as well as all constraints of the operational maintenance restrictions. Using such a model produces infeasible routes as constraints of the operational restrictions are ignored, meanwhile these routes have a poor solution quality as the objective function is overlooked. As opposed to CA, in our proposed algorithm, we start by building the sub-routes, as explains earlier in our algorithm.

As mentioned earlier, these sub-routes are built by including all the possible through connects that maximize. Therefore, the profitability of the generated is already maximized, which in turn results in obtaining better solution quality if compared with CA.

- *Computational time:* the rest of CA procedures focus on conducting two simultaneous tasks; maximizing the profit and satisfying the constraints that described all the operational maintenance restrictions. Using such a way while handling large-scale test instances is very challenging, as it can produce a route with maximized profitability, but the operational maintenance restrictions are violated. In contrary, it is possible to find a route that respects all the operational maintenance restrictions, but the profit is not maximized. This in turn results in producing solutions with poor quality and prolonging the required computational time. In contrast to CA, our algorithm continues its steps by only focusing on one task, which is building complete routes that satisfies the constraints of operational maintenance restrictions. To do so, we connect the generated sub-routes that their profit is already maximized. This results in shortening the computational time significantly.
- *Search mechanism:* A neighbourhood solution is determined in CA by the adoption of a swapping technique, in which the time and place issues, known as the connection feasibility, is only considered. Indeed, this search mechanism does not show good performance for two reasons. Firstly, it cuts two string of aircraft routes, then the tails of the aircraft routes are swapped. Following this part of mechanism can easily generates maintenance infeasible routes. Secondly, cutting the aircraft

routes and swapping the tails results in destroying the through connects, leading to generate solution with minimized profits. These drawbacks are considered while constructing our proposed algorithm, such that the first drawback is avoided by using insertion approaches, as explained in step 6 of the proposed algorithm. The advantage of the forward and backward insertion approaches is that appropriate sub-routes or flights are picked and inserted into the complete route, and on the same time all constraints of operational maintenance restrictions are taken into consideration. Therefore, the constraints of operational maintenance constraints are rare to be violated. The Second drawback of the swapping mechanism is also avoided in the proposed algorithm by using through connects to construct the sub-routes and making these routes fixed, so the breakage of the through connects that results in profit minimization is rare to occur.

Based on the discussion presented in this section, it is clear that the proposed algorithm makes a significant improvement if compared with those results obtained by existing solution methods. Indeed, this result improvements reveal the significance and potential of the proposed algorithm to be applied in airline industry.

3.6.6 Implications on profitability of model 1 and model 2

This section aims to present the implications on profitability after considering the maintenance stations workforce capacity. To do so, we use the proposed solution algorithm and its related modifications to solve model 1 and model 2. The results of this section are briefed in Table 3.6.

Table 3.6: The average solutions obtained from model1 and model 2

Test cases	**Model 1**	**Model 2**	$\%\boldsymbol{IMP}$
	$\bar{Z}_1$	$\bar{Z}_2$	$= (\bar{\boldsymbol{Z}}_1 - \bar{\boldsymbol{Z}}_2)/\,\bar{\boldsymbol{Z}}_1$
Case 1	16,667	16,157	3.06
Case 2	2,333	2,258	3.23
Case 3	5,333	5,151	3.41
Case 4	10,000	9,655	3.45
Case 5	14,909	14,348	3.76
Case 6	21,852	20,950	4.13
Case 7	42,542	40,751	4.21
Case 8	33,899	32,441	4.30
Case 9	59,997	57,321	4.46
Case 10	72,097	68,853	4.50
SIM01	110,425	105,401	4.55
SIM02	140,117	131,486	6.16
SIM03	296,175	270,230	8.76
SIM04	483,533	418,836	13.38
SIM05	1,043,516	847,544	18.78

By looking at the results reported in Table 3.6, it is noticeable that model 1 generates better solution if compared with those generated by model 2. This also is interpreted by the improvement ration (%IMP) that starts from 3.06% in case 1, and steadily increases up to 18.78% in case SIM05. The rationale behind this outperformance lies in that considering workforce capacity helps in avoiding the situation in which more aircraft are scheduled to maintenance stations that suffer from insufficient workforce capacity. Consequently, there is no need to call in additional capacity, resulting in a reduction in the penalty cost, which leads finally to improve the profitability.

3.7 Summary

In this chapter, a new MILP model for OARP is presented, in which all the operational maintenance restriction mandated by FAA, the maintenance stations workforce capacity and the maintenance stations working hours, are taken into consideration. In addition, to solve the presented model, an effective solution algorithm is proposed. Moreover, we modify the proposed model with the aim of assessing implications on profitability after taken into consideration the maintenance stations workforce capacity.

To solve our proposed model, first, a commercial software called CPLEX is adopted. Actually, CPLEX provides optimal solutions for small test cases, but feasible solutions cannot be provided for medium and large test instances. Towards the goal of handling medium and large test instances, the proposed solution algorithm is adopted which shows a good performance while solving different sizes of test instances. For small-scale test instances, the best solutions reach the exact solutions, whereas the average solutions deviate by at most 0.83% from exact solutions. With respect to the computational time, a fast performance for the proposed solution algorithm is shown as it can produce the solution in about 3 seconds, whereas 2.5 hours is taken by CPLEX to solve the same problem. For large-scale test instances, the best solutions provided by the proposed algorithm reach the upper bound, whereas the average solutions deviate by at most 0.66%. from the upper bound. The fast performance of the proposed algorithm is also noticed while solving the large-scale test, as it can solve these cases in a few minutes. It is interesting to mention that the proposed algorithm is tested to solve a test instance that is larger than the size of the largest fleet in the world, named Southwest Airline Boeing 737-700, which include 350 aircraft to cover 3469 flight legs in 4 days. The results show that the best solution reaches the upper bound, whereas the average solution deviates by 0.83%. from the upper bound. These results are achieved within 35 minutes, which is a short computational time.

In this chapter, the experiments are extended for two objectives. The first objective is to benchmark the performance of the proposed solution algorithm with the existing solution methods, like CA. By doing so, the results reveal an outperformance of the proposed algorithm over CA, in different two aspects; the solution quality and the computational time. The second objective is to assess the implication on the profitability after considering the workforce capacity of maintenance stations. The results demonstrate an increase in the profitability by about 18.78% for the largest case after considering the maintenance stations workforce capacity.

Chapter 4 - Joint Optimization using a Leader-follower Stackelberg Game for Coordinated Decision Support System of Stochastic Operational Aircraft Routing and Maintenance Staffing

4.1 Introduction

In this chapter, we discuss the FDARP of airlines along with the MSP of maintenance providers, and our aim is twofold. Firstly, to develop a model for the FDARP of airlines in which the non-propagated delays are reflected in an appropriate way. To achieve this aim, a new scenario-based stochastic framework for the FDARP of airlines is proposed. Secondly, to investigate the inherent interdependence between the FDARP of airlines and the MSP of maintenance providers. For this purpose, we propose a coordinated decision support system for the scenario-based stochastic FDARP and MSP that is formulated as a leader-follower Stackelberg game, in which the scenario-based stochastic FDARP of airlines acts as a leader and the MSP of maintenance providers behaves as a follower. This game model can be modeled as a bi-level optimization model. Towards the goal of solving the proposed bi-level model, we develop a nested ant colony optimization-based algorithm. The viability and the potential of the proposed model are demonstrated by presenting a case study of the proposed model for a major airline and a maintenance provider located in the Middle East.

The remainder of this chapter is organized as follows. In section 4.2, the model description is presented, whereas the bi-level optimization model formulation is described in section 4.3. To solve the proposed bi-level optimization model, a nested ant colony optimization-based algorithm is developed in section 4.4. Using a major airline and a maintenance provider located in the Middle East as a case study, the potential and feasibility of the proposed model is presented in section 4.5. Finally, a summary of the chapter is given in section 4.6.

4.2 Model Description

In this section, the coordinated decision support system of the FDARP of airlines and the MSP of maintenance providers is described. The first part of the system, known as FDARP, is usually formulated in the literature using the expected value of the non-propagated delay. It is important to note here that the flight delay can be categorized into two categories; non-propagated delay and propagated delay. The non-propagated delay can be defined as any delay caused by bad weather, technical problems, airport or maintenance station congestion, etc., which are generalized as on-routing reasons. The propagated delay, on the other hand, is caused due to routing reasons, such that the delay occurs when the aircraft covering the later flight is delayed because of the delay occurred on its previous covered flight. Indeed, the drawback of the expected value approach is that the expected value of some delays is significantly different from the realized value of these delays, owing to the high level of uncertainty associated with the non-propagated delay (Yen and Birge, 2006).

Consequently, using such an approach results in propagating the delay, which in turn increases its related cost. This motivates us to find out different approaches to capture the non-propagated delay. Therefore, in contrast to the expected value approach, we propose a different formulation for FDARP, in which different potential scenarios for the non-propagated delay are studied, so that we can get a suitable look-ahead feature for the delay. Actually, this scenario-based concept results in proposing a scenario-based stochastic framework for FDARP. This framework is proposed because it has been demonstrated as an efficient approach for capturing the stochastic parameters (Chen et al., 2002, Mohammadi et al., 2014, Samimi et al., 2017).

The main task of the developed scenario-based stochastic FDARP of airline is constructing the routing plan, with the objective of minimizing the expected propagated delay cost. To do so, the scenario-based stochastic FDARP of airline uses four things as an input, a set of flight legs, set of flight delay scenarios, a set of aircraft, and a set of aircraft, as shown in the left-hand side of Figure 4.1. Using these fours things as an input to the scenario-based stochastic FADRP of airline results in the generation of the routing plan, as shown in the right-hand side of Figure 4.1. By looking at the routing plan, we can see that each route includes some maintenance visits. To achieve the implementation of the generated routing plan in real practice, two tasks should be performed. Firstly, covering the flight legs, which is the role of airline, and secondly, handling the maintenance visits, which is the responsibility of maintenance providers. Practically, the role of maintenance providers is very crucial, as they are not only responsible for completing the maintenance operations, but also responsible for letting the aircraft depart from the maintenance station punctually. Therefore, the maintenance providers should efficiently manage their workforce capacity by solving the MSP. From this description, we can see that implementing routing plan is a mutual responsibility of the scenario-based stochastic FDARP of airlines and the MSP of maintenance providers, and both problems are related interdependently.

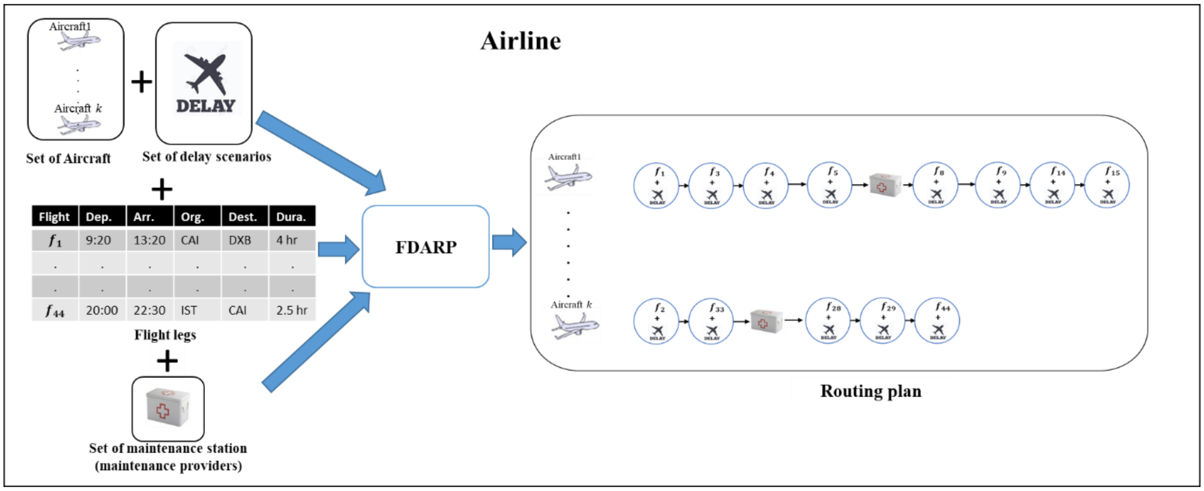


Figure 4.1: Inputs and output of the scenario-based stochastic FDARP

The MSP of maintenance providers, on the other hand, determines the team sizes required to maintain the aircraft of airlines, with the objective of minimizing the labor cost. Towards the goal of arranging these teams, the MSP uses two main things as an input. Firstly, the aircraft to be maintained and their related arrival and departure times, known as the maintenance demand, and secondly, the workforce capacity, as shown left-hand side of Figure 4.1. Using these two things as an input results in the generation of the staffing plan, as shown in the right-hand side of Figure 4.2. Practically, this plan includes group of aircraft and their determined team. Implementing the staffing plan necessities providing the teams, which is the role of maintenance providers, and sending the aircraft to the maintenance stations on time as the responsibility of airlines. If the aircraft fail to arrive on time, the staffing plan will be interrupted, as these aircraft might require more workers to complete the maintenance operation on time. So, implementing staffing plan is again a mutual responsibility of MSP of maintenance providers and scenario-based stochastic FDARP of airlines, and both problems are related interdependently.

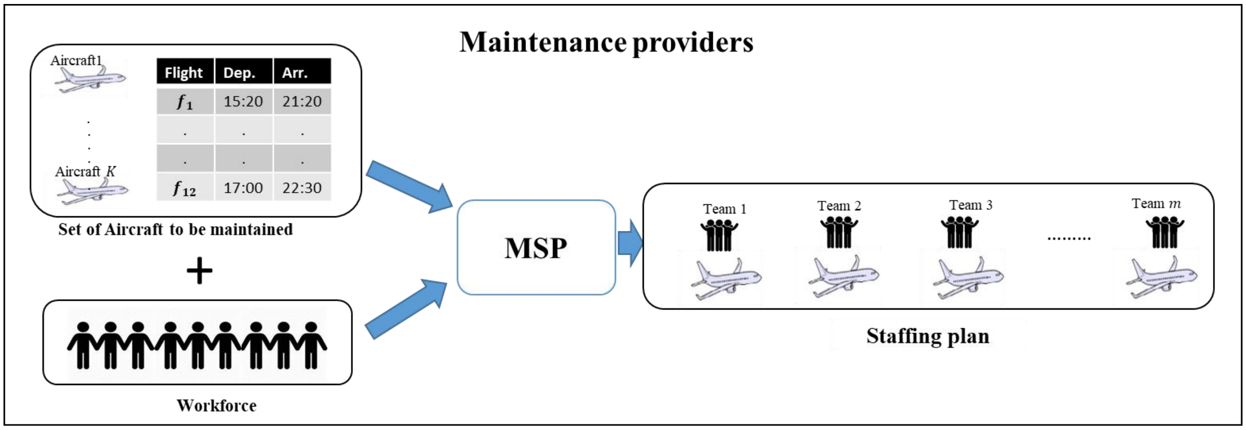


Figure 4.2: Inputs and output of the MSP

The above description reveals that both of scenario-based stochastic FDARP of airlines and MSP of maintenance providers are related independently. Handling this interdependence is much trickier because both the scenario-based stochastic FDARP and MSP represent different business sectors with inconsistent objectives. For airlines, the scenario-based stochastic FDARP is adopted with the objective of minimizing the expected propagated delay cost, which can be achieved by maximizing the team sizes determined by maintenance providers to complete the maintenance operation punctually, which leads to an increase in the labor cost incurred by the maintenance providers. For the maintenance providers, on the other hand, MSP is used with the objective of minimizing the labor cost, which can be achieved by minimizing the team sizes. Consequently, the time taken to complete the maintenance operation for aircraft is prolonged, which in turn results in delay for the next covered flight by the aircraft, leading to an increase in the expected propagated delay cost. In this connection, due to the inconsistent objectives, it is not viable to handle the scenario-based stochastic FDARP and MSP using “all-in-one” approach, in which the objective functions of each problem are combined to form a single objective function. In this sense, modeling scenario-based stochastic FDARP and MSP while taking into account their interdependence and the inconsistent objectives makes the adoption of the coordinated decision support system indispensable. To decide the formulation of the coordinated decision system, it is necessary to precisely investigate the features of scenario-based stochastic FDARP and MSP. Doing so reveals that the scenario-based stochastic FDARP of airlines determines the number of maintenance visits for each aircraft, which constitutes the demand for the maintenance providers. This demand is used as an input for the MSP to construct the staffing plan. Based on these features, it is obvious that the dominating position is held by the scenario-based stochastic FDARP due to determination of the demand, whereas the dominated position is occupied by the MSP as it used the demand determined by the scenario-based stochastic FDARP. These features lead naturally to formulate the coordinated decision support system as a leader-follower Stackelberg game, in which the scenario-based stochastic FDARP behaves as a leader and MSP acts as a follower.

In particular, the scenario-based stochastic FDARP, as a leader, mainly takes decisions related to the number of aircraft to be maintained, and their scheduled arrival and departure times, as shown in Figure 4.3. These decisions are sent to the MSP who behaves as a follower and responds rationally to the received decisions by making decisions regarding the real departure times for the aircraft from maintenance providers. These decisions are sent back to the leader. This process is iterated until reaching to the so-called Stackelberg equilibrium, in which both players are unwilling to change their decisions, as any deviation will result in negative impact in their objective functions. To represent this coordinated decision support system, a bi-level optimization model is adopted, in which the scenario-based stochastic FDARP forms the upper-level, whereas the lower-level is represented by MSP.

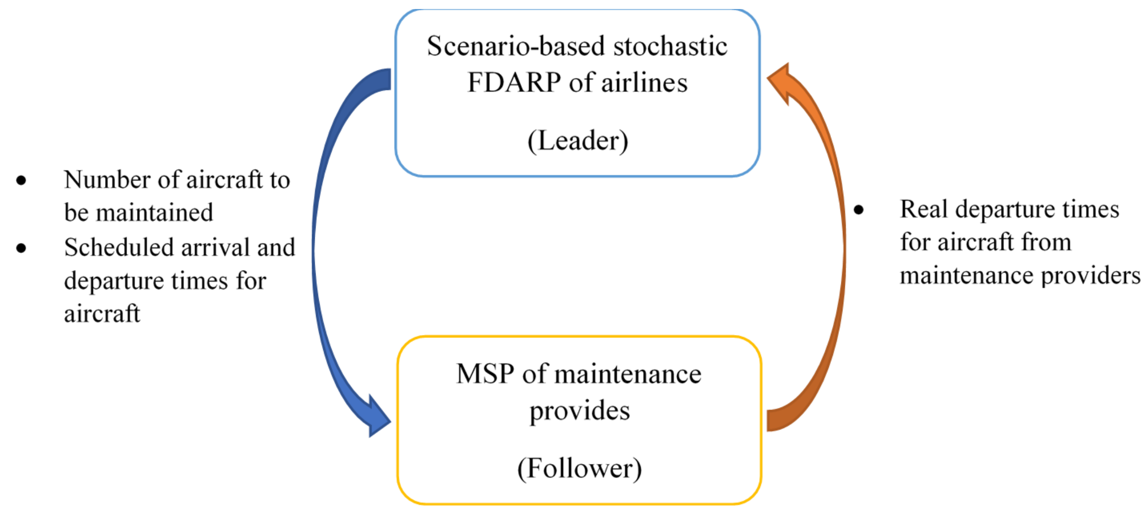


Figure 4.3: Typical representation of leader-follower game model

4.3 Bi-level Optimization Model Formulation

This section mainly proposes the formulation of the bi-level optimization model. Before presenting the model formulation, we first define the model scope and notations.

4.3.1 Scope and notations

The scope of the proposed bi-level optimization model can be described as follows:

- There is a single airline and multiple maintenance providers represented by the proposed model.
- The scenario-based stochastic FDARP of airline is represented by the upper-level part of the bi-level optimization model, whereas the MSP of maintenance providers is represented by the lower-level part of the bi-level optimization model.
- The scope of the scenario-based stochastic FDARP of airline is similar to OARP presented in Chapter 3, in terms of considering a planning horizon of 4-day, taking into account the existing number of maintenance stations, and considering Type A maintenance check.
- The scenario-based stochastic FDARP of airline considers the non-propagated delay of flight leg, by generating different disruption scenarios.
- The maintenance providers focus on solving MSP, while considering deterministic workforce capacity.

After presenting the scope of the proposed bi-level optimization model, we define the notations used throughout the model as follows:

<u>Upper- level optimization model (Scenario-based stochastic FDARP)</u>

<u>Sets</u>

I:	Set of flight legs, indexed by i or j.
MT:	Set of maintenance stations, indexed by m.
K:	Set of aircraft, indexed by k.
A:	Set of airports, indexed by a.

Ψ: Set of scenarios corresponding to non-propagated delay realizations (known as disruption scenarios). This set indexed by ξ.

$v \in \{1,2,\dots,\Omega\}$ Average number of maintenance processes that each aircraft should receive during the planning horizon.

$\{o,t\}$: Artificial source and sink nodes of the aircraft routing network.

Parameters

DT_i: Local time when flight leg i departs from its origin airport, known as departure time.

O_{ia}: Binary indicator for the origin airport of flight leg i. It equals 1 when the origin airport of flight leg i shares a similar location as the airport a, and 0 otherwise.

AT_i: Local time when flight leg i arrives at its destination airport, known as arrival time.

D_{ia}: Binary indicator for the destination airport of flight leg i. It equals 1 when the destination airport of flight leg i shares a similar location as the airport a, and 0 otherwise.

FT_i: Duration of flight leg i.

TRT: Time consumed for getting passengers off, unloading the luggage, changing the gate, boarding, loading the luggage, and fueling the aircraft. It is Known as turn-around time.

T_{max}: Maximum number of allowable flying time since last maintenance process.

C_{max}: Maximum number of allowable take-offs since last maintenance process.

NPD_{ik}^{ξ}: Realized value of the non-propagated delay resulted when flight leg i is covered by aircraft k, while considering disruption scenario ξ.

Mb_{ma}: Binary indicator for maintenance station m. It equals 1 when the location of the maintenance station m and the airport a are identical, and 0 otherwise.

MAT: Duration of Type A maintenance check. The airline assumes this value only in the first round of the coordination between the airline and maintenance companies.

KT: Fleet size.

Ω: Maximum average number of maintenance processes that each aircraft should receive during the planning horizon. It is calculated according to the following equation; $\Omega = \sum_{i \in I} FT_i / (T_{max}\ KT)$

M: A big number.

PD_{ijkv}^{ξ}: Value of propagated delay resulted when aircraft k covers two successive flight legs i and j, before receiving the maintenance process number v, while considering disruption scenario ξ.

PD_{ikv}^{ξ}: Value of propagated delay that is accumulated before covering flight leg i by aircraft k, before receiving maintenance process number v, while considering disruption scenario ξ.

p^{ξ}: Probability in which disruption scenario ξ can be realized.

C_{pD}: Expected cost per each minute of propagated delay.

Decision variables

$x_{ijkv}^{\xi} \in \{0,1\}$: Flight coverage decision variable, while considering disruption scenario ξ. x_{ijkv}^{ξ}=1 when aircraft k covers two successive flight legs i and j, before receiving the maintenance process number v and 0 otherwise.

$y_{imkv}^{\xi} \in \{0,1\}$: Visiting maintenance station decision variable, while considering disruption scenario ξ. y_{imkv}^{ξ}=1 when flight leg i is covered by aircraft k then the aircraft proceeds to maintenance station m to receive the maintenance process number v and 0 otherwise.

$z_{mjkv}^{\xi} \in \{0,1\}$: Leaving maintenance station decision variable, while considering disruption scenario ξ. z_{mjkv}^{ξ}=1 when aircraft k leaves maintenance station m in order to cover flight leg j, after receiving the maintenance process number v and 0 otherwise.

$RTAM_{kv}^{*\xi} > 0$: The ready time an aircraft k completes receiving the maintenance process number v and can resume covering the flight legs, while considering disruption scenario ξ.

Lower-level optimization model (MSP decision model)

Sets

F: Set of flights in which aircraft will be maintained, indexed by f or b.

S: Set of shifts, indexed by s.

$\{o', t'\}$: Artificial starting and ending nodes of the layered graph.

Parameters

SAT_{fm}^{ξ}: Scheduled arrival time of flight f in which aircraft will be maintained, at maintenance station m, while considering disruption scenario ξ received from airline.

SDT_{fm}^{ξ}: Scheduled departure time of flight f in which aircraft will be maintained, at maintenance station m, while considering disruption scenario ξ received from airline.

w_{sm}^{l}: Minimal team size (number of workers) that can be formed to perform a maintenance process, during shift s, at maintenance station m.

w_{sm}^{u}: Maximal team size (number of workers) that can be formed to maintain an aircraft, during shift s, at maintenance station m.

Q_{s}^{max}: Capacity of workforce available during shift s.

l_f: Workload (man-hours) required to maintain the aircraft that covers flight f.

C_{wfsm}: Cost incurred when assigning w workers to maintain an aircraft that covers flight f, during shift s, at maintenance station m.

Decision variables

$wf_{fsm}^{\xi} \in \{w_{sm}^{l}, \dots, w_{sm}^{u}\}$: Number of workers (team size) assigned to maintain an aircraft that covers flight f, during shift s, at maintenance station m, while considering disruption scenario ξ.

$RTAM_{fm}^{\xi} > 0$: Actual ready time for an aircraft that covers flight f to leave the maintenance station m, while considering disruption scenario ξ.

$CC_{fsm}^{\xi} \in \{0,1\}$: =1 if flight f in which aircraft will be maintained, already received the maintenance process, during shift s, at maintenance station m, while considering disruption scenario ξ and 0 otherwise.

4.3.2 Model formulation

After presenting the scope and notations of the proposed bi-level optimization model, we present the detailed formulation of model. As mentioned earlier, the model consists of two levels: upper-level and lower-level.

On the focus of the upper-level part of the bi-level optimization model, it represents the scenario-based stochastic FDARP of airline, which aims to construct the routing plan, with the objective of minimizing the expected propagated delay cost. It should be noted that the scenario-based stochastic FDARP is formulated by following the modified connection network, which is presented in Chapter 3. The nodes and arcs included in the modified connection network ease the process of formulating the scenario-based stochastic FDARP as a multi-commodity network flow-based MILP model, in which each aircraft represent a separate commodity moving throughout the network. To build the routing plan as an output for the scenario-based stochastic FDARP, four different decision variables are used. The first three decision variables are x_{ijkv}^{ξ},y_{imkv}^{ξ} and z_{mjkv}^{ξ}, which represent ordinary arcs, maintenance arcs, and auxiliary arcs, respectively. The fourth decision variable is $RTAM_{kv}^{*\xi}$, which is cast to determine the suitable time for using the auxiliary arcs. The non-propagated delays randomness in the proposed model is represented using a set of generated scenarios Ψ, in which each scenario ξ is associated with the realization of the non-propagated delays of the scheduled flight legs. These scenarios are incorporated in the proposed model to simulate the potential disruptions that may occur in the future, resulting in the generation of a routing plan that can better withstand disruptions and easily be implemented in reality.

With respect to the lower- level part of the bi-level optimization model, it represents the MSP of maintenance providers that its goal is to build the staffing plan and its objective is to minimize the labor cost. It should be mentioned here that MSP is formulated as a layered graph, due to its efficiency to capture the whole feature of staffing and worker allocation problem (Yin and Wang, 2006). This graph consists of three main components: the layers, the nodes and the arcs, as shown in Figure 4.4. For the layers of the graph, they represent the flights in which their covered aircraft require maintenance (known as maintenance operations), besides indicating the starting and ending points. According to the nodes of the graph, each layer consists of number of nodes, which represent the potential team sizes that can be formed to perform the maintenance. Lastly, with respect to the arcs, we incorporate them in the graph with the aim of connecting the layers, so that we can easily apply metaheuristics like ant colony optimization (ACO) as a solution method. To build the staffing plan as an output for the MSP, three different decision variables are used.

Firstly, wf_{fsm} is used in order to help in determining the number of workers required for each maintenance operation. Secondly, $RTAM_{fm}$ is adopted to specify the real departure

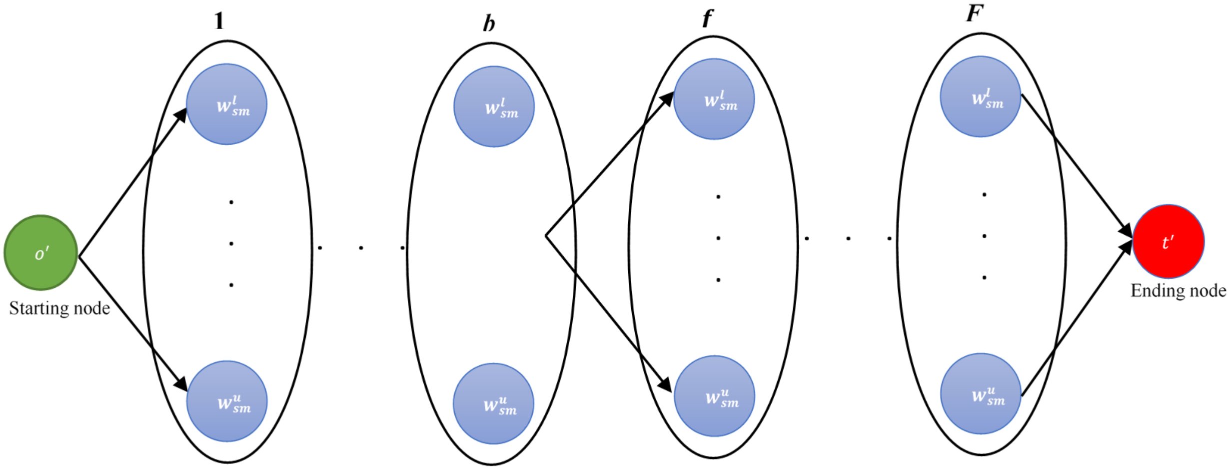


Figure 4.4: Construction of the layered graph

times for the aircraft from the maintenance station. Lastly, CC^{ξ}_{fsm} is incorporated in the model to guarantee each aircraft receives the required maintenance operation.

Based on the predefined notations, the bi-level optimization model can be formulated as follows:

$$min \sum_{\xi \in \Psi} p^{\xi} \left(\sum_{v=1,\dots,\Omega} C_{pD} \left(\sum_{k \in K} \sum_{i \in I} \sum_{j \in I} PD^{\xi}_{ijkv} \, x^{\xi}_{ijkv} \right) \right) \quad (4.1)$$

s.t. $$PD^{\xi}_{ijkv} = PD^{\xi}_{ikv} + (NPD^{\xi}_{ik} - (DT_j - AT_i - TRT))^{+} \quad \forall\, \xi \in \Psi, (i,j) \in I, k \in K, v = 1, \dots, \Omega \quad (4.2)$$

$$\sum_{k \in K} \left(\sum_{j \in I \cup \{t\}} \sum_{v=1,\dots,\Omega} x^{\xi}_{ijkv} + \sum_{m \in MT} \sum_{v=1,\dots,\Omega} y^{\xi}_{imkv} \right) = 1 \qquad \forall\, \xi \in \Psi, i \in I \quad (4.3)$$

$$\sum_{j \in I} x^{\xi}_{ojkv} + \sum_{m \in MT} y^{\xi}_{omkv} = 1 \qquad \forall\, \xi \in \Psi, k \in K, v = 1, \dots, \Omega \quad (4.4)$$

$$\sum_{i\in I} x_{itkv}^{\xi} + \sum_{m\in MT} z_{mtkv}^{\xi} = 1 \qquad \forall\, \xi \in \Psi, k \in K, v = 1, \dots, \Omega \tag{4.5}$$

$$\sum_{j\in I\cup\{o\}} x_{jikv}^{\xi} + \sum_{m\in MT} z_{mikv}^{\xi} = \sum_{j\in I\cup\{t\}} x_{ijkv}^{\xi} + \sum_{m\in MT} y_{imkv}^{\xi} \qquad \forall\, \xi \in \Psi, i \in I, k \in K, v = 1, \dots, \Omega \tag{4.6}$$

$$\sum_{j\in I} \sum_{v=1,\dots,\Omega} y_{jmkv}^{\xi} = \sum_{j\in I\cup\{t\}} \sum_{v=1,\dots,\Omega} z_{mjkv}^{\xi} \qquad \forall\, \xi \in \Psi, m \in MT, k \in K \tag{4.7}$$

$$AT_i + TRT - DT_j \le M(1 - x_{ijkv}^{\xi}) \qquad \forall\, \xi \in \Psi, (i,j) \in I, k \in K, v = 1, \dots, \Omega \tag{4.8}$$

$$\sum_{k\in K} x_{ijkv}^{\xi} \le \sum_{a\in A} D_{ia} O_{ja} \qquad \forall\, \xi \in \Psi, (i,j) \in I, v = 1, \dots, \Omega \tag{4.9}$$

$$\sum_{k\in K} y_{imkv}^{\xi} \le \sum_{a\in A} D_{ia} Mb_{ma} \qquad \forall\, \xi \in \Psi, i \in I, m \in MT, v = 1, \dots, \Omega \tag{4.10}$$

$$\sum_{k\in K} z_{mjkv}^{\xi} \le \sum_{a\in A} Mb_{ma}\, O_{ja} \qquad \forall\, \xi \in \Psi, m \in MT, j \in I, v = 1, \dots, \Omega \tag{4.11}$$

$$RTAM_{kv}^{*\xi} - DT_j \le M\left(1 - z_{mjkv}^{\xi}\right) \qquad \forall\, \xi \in \Psi, m \in MT, j \in I, k \in K, v = 1, \dots, \Omega \tag{4.12}$$

$$RTAM_{kv}^{*\xi} \ge \sum_{i\in I\cup\{o\}} \sum_{j\in I\cup\{t\}} \sum_{m\in MT} (AT_i + MAT) z_{mjkv}^{\xi} \qquad \forall\, \xi \in \Psi, k \in K, v = 1, \dots, \Omega \tag{4.13}$$

$$RTAM_{kv}^{*\xi} \ge \sum_{f\in F} \sum_{j\in I\cup\{t\}} \sum_{m\in MT} RTAM_{fm}^{\xi}\, z_{mjkv}^{\xi} \qquad \forall\, \xi \in \Psi, k \in K, v = 1, \dots, \Omega \tag{4.14}$$

$$\sum_{i\in I\cup\{o\}} \sum_{j\in I} x_{ijkv}^{\xi} \le C_{max} \qquad \forall\, \xi \in \Psi, k \in K, v = 1, \dots, \Omega \tag{4.15}$$

$$\sum_{i\in I\cup\{o\}} \sum_{j\in I} FT_j\ x_{ijkv}^{\xi} \le T_{max} \qquad \forall\, \xi \in \Psi, k \in K, v = 1 \tag{4.16}$$

$$\sum_{i\in I} \sum_{j\in I} FT_j\, x_{ijkv}^{\xi} + \sum_{m\in MT} \sum_{j\in I} FT_j\, z_{mjkv}^{\xi} \le T_{max} \qquad \forall\, \xi \in \Psi, k \in K, v = 2, \dots, \Omega \tag{4.17}$$

$$\sum_{i\in I} \sum_{m\in MT} \sum_{v=1,\dots,\Omega} y_{imkv}^{\xi} \ge 1 \qquad \forall\, \xi \in \Psi, k \in K \tag{4.18}$$

$$x_{ijkv}^{\xi} \in \{0,1\} \qquad \forall\, \xi \in \Psi, (i,j) \in I, k \in K, v = 1, \dots, \Omega \tag{4.19}$$

$$y_{imkv}^{\xi} \in \{0,1\} \qquad \forall\, \xi \in \Psi, i \in I, m \in MT, k \in K, v = 1, \dots, \Omega \tag{4.20}$$

$$z_{mjkv}^{\xi} \in \{0,1\} \qquad \forall\, \xi \in \Psi, m \in MT,\ j \in I, k \in K, v = 1, \dots, \Omega \tag{4.21}$$

$$RTAM_{kv}^{*\xi} > 0 \qquad \forall\, \xi \in \Psi, k \in K, v = 1, \dots, \Omega \tag{4.22}$$

where given decision variables (y_{imkv}^{ξ}, z_{mjkv}^{ξ} and $RTAM_{kv}^{*\xi}$) are used for solving:

$$min \sum_{m\in MT} \sum_{s\in S} \sum_{f\in F} C_{wfsm}\, wf_{fsm}^{\xi} CC_{fsm}^{\xi} \tag{4.23}$$

s.t. $$RTAM_{fm}^{\xi} \geq SDT_{fm}^{\xi} + \left(SAT_{fm}^{\xi} + TRT + l_f/wf_{fsm}^{\xi} - SDT_{fm}^{\xi}\right) \quad \forall\, \xi \in \Psi, f \in F, m \in MT \tag{4.24}$$

$$\sum_{f\in F} wf_{fsm}^{\xi} \leq Q_s^{max} \qquad \forall\, \xi \in \Psi, s \in S, m \in MT \tag{4.25}$$

$$\sum_{s\in S} \sum_{m\in MT} CC_{fsm}^{\xi} = 1 \qquad \forall\, \xi \in \Psi, f \in F \tag{4.26}$$

$$SAT_{fm}^{\xi} = \sum_{k\in K} \sum_{i\in I} \sum_{v\in V} AT_i\, y_{imkv}^{\xi} CC_{fsm}^{\xi} \qquad \forall\, \xi \in \Psi, f \in F, s \in S, m \in MT \tag{4.27}$$

$$SDT_{fm}^{\xi} = \sum_{k\in K} \sum_{j\in I} \sum_{v\in V} RTAM_{kv}^{*\xi}\, z_{mjkv}^{\xi}\, CC_{fsm}^{\xi} \qquad \forall\, \xi \in \Psi, f \in F, s \in S, m \in MT \tag{4.28}$$

$$wf_{fsm}^{\xi} \in \{w_{sm}^{l}, \dots, w_{sm}^{u}\} \qquad \forall\, \xi \in \Psi, f \in F, s \in S, m \in MT \tag{4.29}$$

$$RTAM_{fm}^{\xi} > 0 \qquad \forall\, \xi \in \Psi, f \in F, m \in MT \tag{4.30}$$

$$CC_{fsm}^{\xi} \in \{0,1\} \qquad \forall\, \xi \in \Psi, f \in F, s \in S, m \in MT \tag{4.31}$$

The scenario-based stochastic FDARP, as the upper-level part of the bi-level optimization model, is represented by Eqs. (4.1) - (4.22), whereas the MSP, as the lower-level part of the bi-level optimization model, is illustrated by Eqs. (4.23) - (4.31).

On the focus of the upper-level part of the bi-level optimization model, the objective function is to minimize the expected propagated delay cost, as stated in Eq. (4.1). Constraints (4.2) describe the calculation of the propagated delay. It should be noted that the rest of the constraints that are applied to the upper-level are similar to those applied to the OARP, which is presented in Chapter 3, except ready time constraints in Eqs. (4.13) and (4.14). Initially, in the first round of the coordination between the scenario-based stochastic FDARP and MSP, $RTAM_{kv}^{*\xi}$ is determined according to constraints (4.13), in which the airline assume the Type A maintenance duration. In real practice, it is not viable for airline to assume the Type A maintenance duration, as this duration should be calculated by maintenance providers. In this connection, in the subsequent rounds of coordination between the scenario-based stochastic FDARP and MSP, constraints (4.13) becomes redundant, and $RTAM_{kv}^{*\xi}$ is calculated by following constraints (4.14). These constraints guarantee that the routing plan is built according to the actual ready time of aircraft determined by the MSP of maintenance providers. In this model, constraints (4.14) form the linkage between the upper and lower-levels of the bi-level optimization model.

Moving to the lower-level part of the bi-level optimization model, the objective function is to minimize the total labor cost incurred by maintenance providers, as represented by Eq. (4.23). Constraints (4.24) describe how the maintenance providers calculate the actual ready time when the aircraft can leave the maintenance providers. Towards the goal of constructing a feasible staffing plan, it is important to consider the worker capacity in each shift. Therefore, constraints (4.25) are cast, to guarantee that the workforce capacity in each shift is violated by the total number of workers allocated to serve the aircraft. Constraints (4.26) are formulated to ensure that during the planning horizon, each aircraft receives the maintenance service exactly once.

Since the MSP acts as a follower of the leader-follower Stackelberg game, some information should be received from the leader. For this purpose, constraints (4.27) and (4.28) are formulated. Constraints (4.27) and (4.28) help in calculating the scheduled arrival and departure times for each aircraft, respectively. These two constraints are cast according to decision variables y^{ξ}_{imkv}, z^{ξ}_{mjkv} and $RTAM^{*\xi}_{kv}$ that are received from the leader. Finally, constraints (4.29) – (4.31) indicate the domain definition of the decision variables.

To summarize, the bi-level optimization model starts with the scenario-based stochastic FDARP as the upper-level, which constructs the routing plan, including covering the flight legs by the help of decision variable x^{ξ}_{ijkv}, and preparing maintenance visits to the aircraft through decision variables y^{ξ}_{imkv}, z^{ξ}_{mjkv} and $RTAM^{*\xi}_{kv}$. The rouging plan (referred to as y^{ξ}_{imkv}, z^{ξ}_{mjkv} and $RTAM^{*\xi}_{kv}$) is sent to the MSP, as the lower-level, which construct the staffing plan by the help of decision variables wf_{fsm}, $RTAM_{fm}$ and CC^{ξ}_{fsm}. The staffing plan (referred to as $RTAM_{fm}$) is sent back to the upper-level. If the routing plan is interrupted, the scenario-based stochastic FDARP is resolved and the new routing plan is sent to the lower-level. This process is iterated until finding out the Stackelberg equilibrium, in which both levels are not willing to change their decisions, as any change causes a negative impact to the objective function.

4.4 Solution Method

To get the Stackelberg equilibrium, it is necessary to solve the proposed bi-level optimization model. Before presenting our solution method, we discuss briefly the existing solution methods. The bi-level optimization model can be solved by using two different approaches: indirect and direct. To use the indirect approach, it is necessary to change the structure of the model from a bi-level to a single level, then solve the single level by adoption of some methods like B&B based on K times best method (Bard and Falk, 1982), Karushe-Kuhne-Tucker (KTT) conditions method (Fortuny-Amat et al., 1981) and penalty function method (Anandalingam and White, 1990). Solving our bi-level optimization model by following the indirect approach is not promising for two reasons. Firstly, it ignores some features of the model, like each level represents a different company with a self-interest goal. Secondly, different representation for the proposed model might be appeared, as the leader's decision power might be dominated by the follower's decision when converting the bi-level to a single level. The direct approach, on the other hand, aims at solving the bi-level model in a direst way using some methods such as satisfactory solution method (Muñoz and Abdelaziz, 2012). The advantage of this approach is respecting the structure of the bi-level model. However, it becomes quite challenging for the direct solution methods to solve large-scale network optimization problems. This appears when the efficiency of the direct solution methods is significantly reduced because the network model includes a large number of nodes (Wang et al., 2016). Due to the fact that each level of our bi-level model (known as scenario-based stochastic FDARP and MSP) belongs to large scale network optimization problems, thus, solving the proposed bi-level optimization model using the direct approach is challenging.

It is noteworthy that each level of the proposed bi-level model is NP-hard (Sarac et al., 2006, Yin and Wang, 2006). In the light of this fact, solving the proposed bi-level optimization model by adoption of metaheuristics is reasonable. This is because the metaheuristics have shown successful applications while solving different problems, such as crew scheduling problem (Ozdemir and Mohan, 2001), vehicle routing problem (Wang et al., 2015), aircrew rostering problem (Lučic and Teodorovic, 1999), and control system problems (Hashim et al., 2015, Hashim and Abido, 2015, Hashim et al., 2016). As described earlier, each of scenario-based stochastic FDARP and MSP are modeled as network-based problems, for which ACO has shown successful application while solving large and complex network-based problems (Huang et al., 2018, Mahato et al., 2017, Skinderowicz, 2017, Balseiro et al., 2011). All the previous observations motivate us to propose a nested ACO based-algorithm to solve the bi-level model. This algorithm consists of two levels; upper-level ACO-based algorithm for the scenario-based stochastic FDARP and the lower-level ACO-based algorithm for MSP. These two levels of the nested ACO-based algorithm are designed as each level of the proposed bi-level optimization model has its distinctive features and goals.

4.4.1 Upper-level ACO-based algorithm

In this section, we describe the main two steps that forms the upper-level ACO-based algorithm. These steps can be described as follows:

1) Route construction. This step is conducted through the ant (i.e. each ant simulates an aircraft $k \in K$), by scouting throughout the network. Each ant constructs its own route, which starts from the source node and ends at the sink node. Between the starting and ending nodes, the ant looks for covering flight legs. Suppose that an ant currently covers flight leg represented by node i and looks for covering its next flight leg represented by node j. To select the next flight leg, we apply the following state transition rule:

$$j = \begin{cases} \arg\_max_{j \in N_i^k} \left\{ \left[{\tau_{ij}}^{\xi}\right]^{\alpha} \left[{\eta_{ij}}^{\xi}\right]^{\beta} \right\} & if\ q \leq q_0 \\ J & if\ q > q_0 \end{cases} \tag{4.32}$$

where N_i^k denotes the group of possible flight legs that ant k can select after covering flight leg i. The term ${\tau_{ij}}^{\xi}$ represent the pheromone trail of the ordinary arc $ord(i, j)$ that results during consideration of the disruption scenario ξ. On the other hand, ${\eta_{ij}}^{\xi}$ is the heuristic function of the same arc, which is equal to $1/(C_{pD} * PD_{ijkv}^{\xi})$. The relative importance of ${\tau_{ij}}^{\xi}$ and ${\eta_{ij}}^{\xi}$ is denoted by parameters α and β, respectively. q_0 is the exploration threshold parameter ($0 \leq q_0 \leq 1$) and q represents random number that is generated according to the uniform distribution $[0 \sim 1]$. Ideally, the value q guides the ant to select the next flight leg j. In case of $q \leq q_0$, the ant selects flight leg j which its arc $ord(i, j)$ carries the greatest ${\tau_{ij}}^{\xi}$ and ${\eta_{ij}}^{\xi}$. Conversely, in the case of $q > q_0$, the flight leg j is selected using the probability rule below:

$$P_{ij}^k = \frac{\left[{\tau_{ij}}^{\xi}\right]^{\alpha}\left[{\eta_{ij}}^{\xi}\right]^{\beta}}{\sum_{j \in N_i^k}\left[{\tau_{ij}}^{\xi}\right]^{\alpha}\left[{\eta_{ij}}^{\xi}\right]^{\beta}} \qquad if\ j \in N_i^k \tag{4.33}$$

This selection part continues until all the flight legs are covered by the aircraft.

2) Updating the pheromone trail. This step is considered the most critical part of ACO, in which the pheromone trail is updated with the aim of reflecting the quality of the obtained solution. Taking ρ as the evaporation rate parameter ($0 < \rho < 1$), the pheromone trail can be updated in accordance with the following rule:

$$\tau_{ij,new}^{\xi} = (1 - \rho)\tau_{ij,old}^{\xi} + \Delta\, {\tau_{ij}}^{\xi} \tag{4.34}$$

For each iteration, a uniform reduction of the pheromones can be achieved using the the first term $(1-\rho)\tau_{ij,old}^{\xi}$ of Eq. (4.34). The advantage of this phermone uniform reduction is that in the next iteration, the ants can ignore the bad routes and look for better routes. In a disruption scenario ξ, the pheromone quantity on the edge $ord(i,j)$ is defined by the second term $\Delta\,\tau_{ij}{}^{\xi}$, which is only applied to update the phermone trails of the edges that constitute the best solution found so far. Thus, such updating enables the ants to look for better routes in the subsequent iterations. By taking Q as the control factor in depositing the pheromone, then $\Delta\,\tau_{ij}{}^{\xi}$ is calculated using Eq. (4.35).

$$\Delta\,\tau_{ij}{}^{\xi} = \frac{Q}{cost\left(A_{best}^{\xi}\right)} \qquad if\{i,j\} \subseteq A_{best}^{\xi} \tag{4.35}$$

The value of Q specifies either a local optimal convergence or a random search for the algorithm. The $cost\left(A_{best}^{\xi}\right)$ represents the lowest propagated delay cost between the start and the present iteration, and ξ is the handling disruption scenario.

4.4.2 Lower-level ACO-based algorithm

Similar to the previous section, in this section the main two steps of the lower-level ACO-based algorithm is described as follows:

1) Staffing construction. This step is almost similar to the first step of the upper-level ACO-based algorithm, as it is performed by the help of ants. Each ant moves throughout the layered graph to build its own path by starting from the source node, visiting each layer (i.e. each layer represents a maintenance process) in sequence,

and eventually ending at the sink node. When the ants visit each layer, they select a node with an appropriate team size. To clarify the step of selecting the appropriate team size, suppose an ant currently visits layer b and the next layer f contains the team size to be selected. This selection is applied according to the following state transition rule:

$$w = \begin{cases} \arg\_max_{w \in N_f'^{ant}} \left\{ \left[\tau_{bfw}^{\xi}\right]^{\alpha'} \left[\eta_{bfw,worker}^{\xi}\right]^{\beta'} \right\} & if\ q \leq q_0' \\ W & if\ q > q_0' \end{cases} \tag{4.36}$$

where $N_f'^{ant}$ is the set of potential team sizes that can be determined to maintain flight f. The term τ_{bfw}^{ξ} is the pheromone trail of the edge connecting layers b and f, such that layer f will be served by team size w, under disruption scenario ξ. On the other hand, $\eta_{bfw,worker}^{\xi}$ is the heuristic function for the same edge, which is equal to $1/(wf_{fsm}^{\xi} * C_{wfsm})$. The two parameters $\alpha'$and β'represent the relative importance of τ_{bfw}^{ξ} and $\eta_{bfw,worker}^{\xi}$, respectively. q_0' is the exploration threshold parameter $(0 \leq q_0' \leq 1)$ and q represents random number that is generated according to the uniform distribution $[0{\sim}1]$. Ideally, value q determines a team size to be selected by the ant. In the case of $q \leq q_0'$, a team size w in which its edge has the best information heuristic and pheromone trail is selected. Conversely, if $q > q_0'$, the ant chooses a team size w according to the following probability rule:

$$P_{bfw}^{ant} = \frac{\left[\tau_{bfw}^{\xi}\right]^{\alpha'} \left[\eta_{bfw,worker}^{\xi}\right]^{\beta'}}{\sum_{l \in N_f'^{ant}} \left[\tau_{bfw}^{\xi}\right]^{\alpha'} \left[\eta_{bfw,worker}^{\xi}\right]^{\beta'}} \qquad if\ w \in N_f'^{ant} \tag{4.37}$$

2) Updating the pheromone trail: This process is done using Eq. (4.38)

$$\tau_{bfw,new}^{\xi} = (1-\rho')\tau_{bfw,old}^{\xi} + \Delta\,\tau_{bfw}{}^{\xi} \tag{4.38}$$

Similar to the updaing phermone trail of the upper-level ACO-based algorithm, the first term $(1-\rho')\tau_{bwf,old}^{\xi}$ is adopted after each ant completes its path, whereas the second term $\Delta\,\tau_{bfw}{}^{\xi}$ is utilized to update the the phermone trails of the edges that form the best solution found so far. $\Delta\,\tau_{bfw}{}^{\xi}$ represents the pheromone quantity on the edge connected between flights b and f, such that flight f is served by team size of w, under disruption scenario ξ. $\Delta\,\tau_{ij}{}^{\xi}$ can be determined by using the following rule:

$$\Delta\,\tau_{bfw}{}^{\xi} = \frac{Q'}{cost\left(B_{wm,best}^{\xi}\right)} \qquad if\ edge \subseteq B_{wm,best}^{\xi} \tag{4.39}$$

where Q' is the factor controlling the process of pheromone depositing. The $cost\left(B_{wm,best}^{\xi}\right)$ denotes the lowest cost achieved from the start to the present iteration, while determining the staffing plan for maintenance station m, under disruption scenario ξ.

4.4.3 Nested ACO based-algorithm for bi-level optimization model

Solving the bi-level optimization model that captures the interdependence between the scenario-based stochastic FDARP and MSP requires an algorithm that can capture this interdependence. For this purpose, a nested ACO-based algorithm is proposed. In particular, the nested algorithm starts with the upper-level ACO-based algorithm in order to solve the scenario-based stochastic FDARP and provide the routing plan, including covering the flight legs by the help of decision variable x_{ijkv}^{ξ}, and preparing maintenance visits to the aircraft through decision variables y_{imkv}^{ξ}, z_{mjkv}^{ξ} and $RTAM_{kv}^{*\xi}$. The rouging plan (referred to as y_{imkv}^{ξ}, z_{mjkv}^{ξ} and $RTAM_{kv}^{*\xi}$) is sent to the lower-level ACO-based algorithm as an input to solve the MSP and provide the staffing plan by the help of decision variables wf_{fsm}, $RTAM_{fm}$ and CC_{fsm}^{ξ}. All the lower-level best solutions are relayed to the upper-level ACO-based algorithm to be re-run and its solution adjusted. The algorithm undergoes some iterations until finding out the Stackelberg game equilibrium. The following is a detailed stepwise procedure for implementing the nested ACO-based algorithm:

Step 1.0: Set the initial value for the nested ACO-based algorithm parameters including; ($\alpha, \beta, q_0, \rho, Q, \alpha', \beta', q_0', \rho', Q'$, number of ants for each ACO).

Step 1.1: Generate a specific number for the disruption scenarios. Next, put the generated scenarios in a list called (Ψ).

Step 1.2: Calculate the value of the maximum average number of maintenance processes that each aircraft should receive, known as Ω. Then, set the value for the maximum number of iterations.

Step 1.3: For the number of iterations, initialize it to be one.

Step 1.4: Construct the routing plan by applying the upper-level ACO-based algorithm that is described in the following Steps 1.5 - 1.13:

Step 1.5: Examine the condition of the Ψ list. In the case of nonempty Ψ list, proceed to Step 1.6, otherwise move to step 1.12.

Step 1.6: Select a single disruption scenario ξ from the Ψ list.

Step 1.7: Construct two lists such that the first one includes all the aircraft operated by the airline (K), whereas the second one stores all the flight leg nodes that should be covered by the aircraft (I).

Step 1.8: Construct the aircraft routes by following the next sub-steps:

Step a: Using the K list, examine its condition. In the case of empty K list, proceed to Step 1.9, otherwise pick randomly a single ant or aircraft k and place it on the source node of the modified connection network.

Step b: By using the coverage constraints in Eq. (4.3), examine the status of the I list. In case of non-empty I list, proceed to Step c, else move to step 1.9.

Step c: By considering the route initiation constraints defined in Eq. (4.4), let ant k starts its route by covering a flight leg i from the I list.

Step d: By taking into consideration the time and place constraints stated in Eqs. (4.8) and (4.9), scan through the I list to identify the possible flight legs to be covered. In case of no more possible flights to be covered, proceed to Step j, else move to Step e.

Step e: By adoption of the state transition equation expressed in Eqs. (4.32) and (4.33), select the next flight leg j among the possible flight legs to be covered.

Step f: Using the constraints given in Eqs. (4.15) - (4.18), check whether the operational maintenance restrictions are violated or not because of selecting flight leg j. In case of violation, proceed to Step g, else move to Step i.

Step g: By taking into consideration the place constraints for maintenance stations shown in Eq. (4.10), prepare a maintenance visit for the aircraft.

Step h: After completing the maintenance operation, the aircraft should resume covering the flight legs by following these constraints (Eqs. 4.6, 4.7, and 4.11 - 4.14). Note that when the number of iterations =1, the departure time for the aircraft from the maintenance station is assumed by scenario-based stochastic FDARP of airline by following the constraints in Eq. (4.13), but for the remaining iterations, this time is determined by the MSP of maintenance providers by following the constraints in Eq. (4.14), which contain the decisions ($RTAM_{fm}^{\xi}$) stored in Step 2.8.

Step i: Insert the selected flight leg j to the constructed route before excluding it from the I list. Next, move to step d to repeat the process of selecting another flight leg.

Step j: Terminate the route for the aircraft k by placing it in the sink node according to the route completion constraints shown in Eq. (4.5).

Step k: Exclude aircraft k from the K list, then proceed to step a.

Step 1.9: keep the pheromone trails up-to-date for all arcs of the obtained solution by using Eqs. (4.34) and (4.35).

Step 1.10: For the existing iteration, calculate the solution regarding the disruption scenario ξ (Z_{iter}^{ξ}). In the case of obtaining a better solution, update the best solution for the disruption scenario ξ (Z_{best}^{ξ}).

Step 1.11: Save the best solution of the present disruption scenario Z_{best}^{ξ} and the decision variables (y_{imkv}^{ξ}, z_{mjkv}^{ξ} and $RTAM_{kv}^{*\xi}$) that are related to it. Exclude ξ from the Ψ list before moving to Step 1.5.

Step 1.12: Augment the best solution gotten from each disruption scenario ($Z_{iter,routing} = \sum_{\xi\in\Psi} Z^{\xi}_{best} * p^{\xi}$) in order to assess the solution of the present iteration.

Step 1.13: Check whether the stopping criteria for the upper-level ACO (SAC-UL) is satisfied or not, given the satisfaction status of the lower-level ACO stopping criteria (SAC-LL). There are three possibilities for this step, as follows:

- If SAC-UL is satisfied or not, while SAC-LL is not satisfied, then go to Step 1.14.
- If SAC-UL is not satisfied, while SAC-LL is satisfied, then use the same list generated in Step 1.1 to update the Ψ list, increase the number of iterations, and proceed to Step 1.4.
- If both SAC-UL and SAC-LL are satisfied, then go to Step 2.11.

Step 1.14: Construct the staffing plan by applying the lower-level ACO-based algorithm that is described in the following Steps 2.1 – 2.10:

Step 2.1: Design a routing list (RL^{ξ}) to receive the routing solution under different disruption scenarios received from Step 1.11. For each routing solution, design a maintenance station list (MT^{ξ}) that contains the maintenance stations that will be visited by the aircraft.

Step 2.2: Examine the status of the RL^{ξ}_m list. In case of nonempty RL^{ξ}_m list, proceed to Step 2.3, otherwise move to Step 2.9.

Step 2.3: Select a routing solution rl^{ξ}_m from the RL^{ξ}_m list.

Step 2.4: Examine the condition of the MT^{ξ} list. In case of nonempty MT^{ξ} list, proceed to Step 2.5, else move to Step 2.7.

Step 2.5: Pick a maintenance station m^{ξ} from the MT^{ξ} list. Based on the routing solution rl_{m}^{ξ}, extract the flights in which aircraft will be maintained at the maintenance station m^{ξ}, and store them in a list called (F).

Step 2.6: Determine team sizes for each aircraft by executing these sub-steps:

Step a: Construct the ANT list.

Step b: Pick randomly an ant from the ANT list and put it in the starting node of the layered graph as its current position for the ant.

Step c: Check the status of the F list. If all flights stored in the F list are visited by the picked ant., proceed to Step i, otherwise move to Step d.

Step d: Using the F list, pick the first unvisited f flight and mark the picked flight's layer as a next position to be visited by the ant.

Step e: For the picked flight, calculate its scheduled arrival and departure time by following the constraints stated in Eqs. (4.27) and (4.28).

Step f: By using results of Step e, determine the targeted shift. Then, determine the possible team sizes by taking the constraints stated in Eqs. (4.25), (4.26), (4.29), and (4.31) into consideration.

Step g: By adoption of the state transition rule expressed in Eqs. (4.36) and (4.37), determine the team size required to maintain the picked flight.

Step h: Modify the status of the flight f to be a visited flight in the F list. Next, Put the ant on the flight f layer, mark this position as its current position for the next selection and go to Step c.

Step i: Terminate the ant path by putting it on the ending node of the layered graph.

Step j: Use the rule represented in Eqs. (4.38) and (4.39) to keep the pheromone trails up-to-date.

Step k: Examine the condition of the ANT list. In case of nonempty ANT list, proceed to Step l, otherwise move to step m.

Step l: Update all the flights in F list to be unvisited flights, then move to Step c.

Step m: Assess the solution of the present iteration (Z_m^{ξ}), update the best solution ($Z_{m,best}^{\xi}$) if needed, and proceed to Step 2.4.

Step 2.7: Compute the best solution for the staffing plan under disruption scenario ξ, calculated as $Z_{total,m,best}^{\xi} = \sum_{m \in MT} Z_{m,best}^{\xi}$. Then, proceed to Step 2.2.

Step 2.8: Store the best solution of each disruption scenario ξ, and its related decision variables ($RTAM_{fm}^{\xi}$).

Step 2.9: Assess the solution of the present iteration $Z_{iter,staffing} = \sum_{\xi \in \Psi} Z_{total,m,best}^{\xi} * p^{\xi}$.

Step 2.10: Check whether the stopping criteria for the lower-level ACO (SAC-LL) is satisfied or not, given the satisfaction status of the upper-level ACO stopping criteria (SAC-UL). There are three possibilities for this step, as follows:

- If SAC-LL is satisfied or not, while SAC-UL is not satisfied, then use the same list generated in Step 1.1 to update the Ψ list, increase the number of iterations, and proceed to Step 1.4.
- If SAC-LL is not satisfied while SAC-UL is satisfied, then increase the number of iterations, and proceed to Step 1.14.
- If both SAC-LL and SAC-UL are satisfied, then proceed to Step 2.11.

Step 2.11: Return the best solution generated by both levels. Since these solutions generate the status in which both players are unwilling to adjust their decisions, the Stackelberg equilibrium is derived, and we terminate the algorithm.

Figure 4.5 presents the flow chart of the nested ACO-based algorithm. As mentioned earlier, this algorithm consists of two main parts; the upper-level ACO-based algorithm, as shown in the left-hand side of the figure, and the lower-level ACO-based algorithm as shown in the right-hand side of the figure. For the sake of computational convenience, we set the stopping criterion for each level of the nested ACO-based algorithm to be the convergence (i.e. 100 successive iterations without solution improvement), or when completing the maximum number of iterations (i.e. 500 iterations), whichever comes first. If the stopping criteria for both levels of the nested ACO-based algorithm are satisfied, then the nested algorithm is terminated, and the Stackelberg equilibrium is derived.

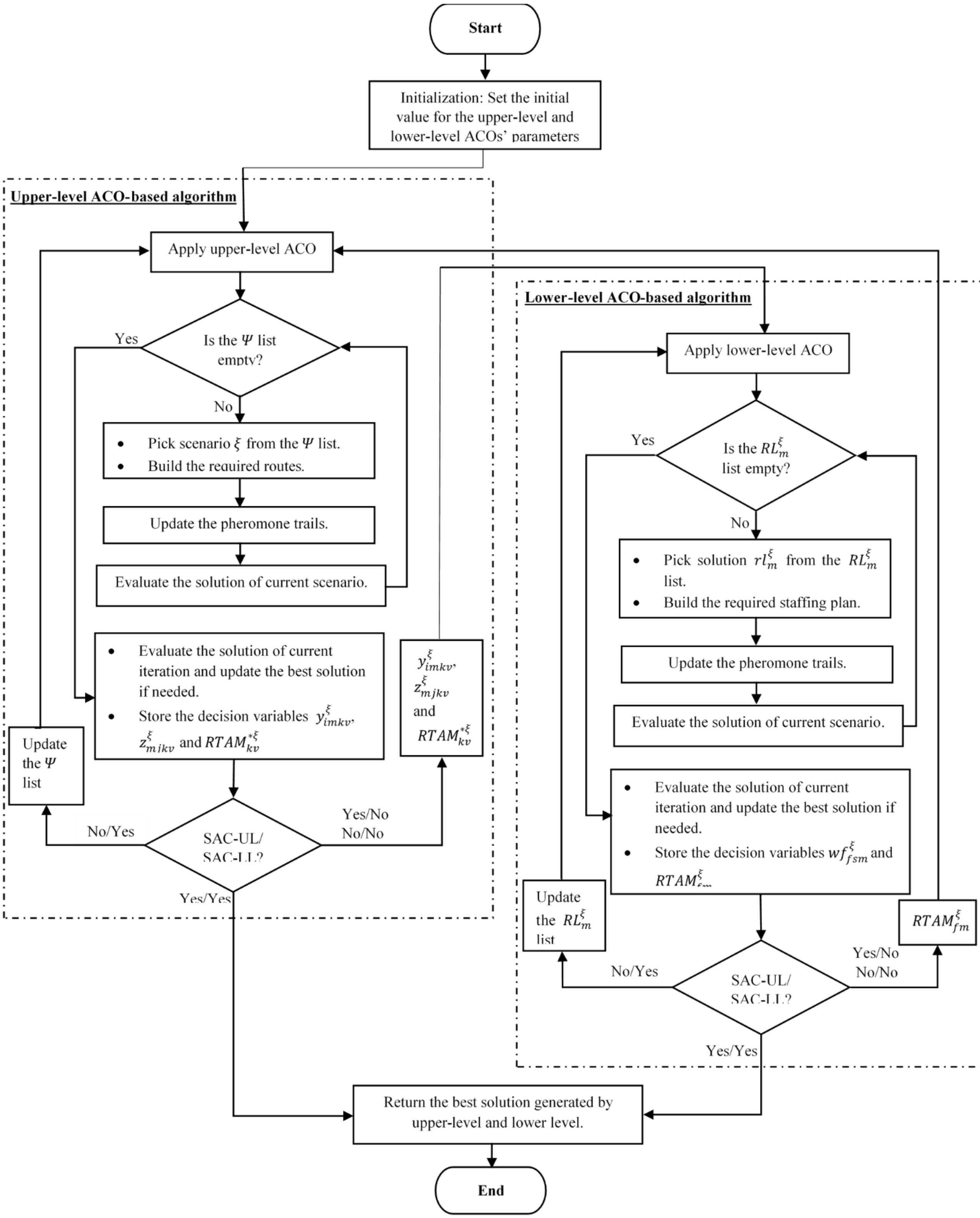


Figure 4.5: Flow chart of the bi-level nested ACO algorithm

4.5 Case Study

4.5.1 Data setting

After developing a bi-level optimization model that allows coordination among the decision makers of the scenario-based stochastic FDARP and MSP, it is necessary to demonstrate the effectiveness of the proposed model. Towards this goal, we present a case study for a Middle Eastern major airline and four maintenance providers. The detailed information regarding the collected data is presented in Table 4.1. Note that the proposed model and algorithm were coded in MATLAB R2014a, and tested using a laptop running on Windows 10 operating system, with a processor of Intel i7 2.50 GHz and 8 GB of RAM memory.

Table 4.1: Characteristics of the collected data

Airline	
I	320 flight legs
FS	30 aircraft
A	8 airports
C_{max}	10 take-offs
T_{max}	40 hours
MT	4 maintenance providers
TRT	45 minutes
MAT	8 hours
C_{pD}	$C_{pD} = \begin{cases} 750 & PD_{ijkv}^{\xi} \leq 15\ minutes \\ 1250 & PD_{ijkv}^{\xi} > 16\ minutes \end{cases}$

Maintenance provider 1		Maintenance provider 2		Maintenance provider 3		Maintenance provider 4	
• l_f=50 hours		• l_f=50 hours		• l_f=50 hours		• l_f=50 hours	
Morning shift		Morning shift		Morning shift		Morning shift	
w_{Sm}^{l}	8 workers	w_{Sm}^{l}	6 workers	w_{Sm}^{l}	8 workers	w_{Sm}^{l}	7 workers
w_{sm}^{u}	15 workers	w_{sm}^{u}	12 workers	w_{sm}^{u}	15 workers	w_{sm}^{u}	12 workers
Q_s^{max}	150 workers	Q_s^{max}	120 workers	Q_s^{max}	100 workers	Q_s^{max}	80 workers
wf_{fsm}	CC_{fsm}	wf_{fsm}	CC_{fsm}	wf_{fsm}	CC_{fsm}	wf_{fsm}	CC_{fsm}
8	7200	6	6000	8	7500	7	8000
9	8000	7	7000	9	8500	8	9000
10	9500	8	7900	10	10000	9	9500
11	10000	9	8750	11	10300	10	10700
12	10500	10	10000	12	10700	11	11000
13	11200	11	10500	13	11600	12	12000

14	11800
15	12500

Afternoon shift	
w_{sm}^{l}	5 workers
w_{sm}^{u}	10 workers
Q_s^{max}	100 workers
wf_{fsm}	CC_{fsm}
5	6700
6	7300
7	8000
8	8700
9	9500
10	10200

Night shift	
w_{sm}^{l}	2 workers
w_{sm}^{u}	5 workers
Q_s^{max}	50 workers
wf_{fsm}	CC_{fsm}
2	6700
3	7200
4	7900
5	8300

12	10900

Afternoon shift	
w_{sm}^{l}	4 workers
w_{sm}^{u}	8 workers
Q_s^{max}	80 workers
wf_{fsm}	CC_{fsm}
4	6250
5	7000
6	7850
7	8500
8	9200

Night shift	
w_{sm}^{l}	2 workers
w_{sm}^{u}	4 workers
Q_s^{max}	40 workers
wf_{fsm}	CC_{fsm}
2	7500
3	8000
4	8400

14	12400
15	13000

Night shift	
w_{sm}^{l}	5 workers
w_{sm}^{u}	10 workers
Q_s^{max}	100 workers
wf_{fsm}	CC_{fsm}
5	8500
6	8900
7	9500
8	10700
9	11200
10	11800

Night shift	
w_{sm}^{l}	5 workers
w_{sm}^{u}	10 workers
Q_s^{max}	80 workers
wf_{fsm}	CC_{fsm}
5	9000
6	9700
7	10500
8	11250
9	12000
10	13200

4.5.2 Scenario generation

One of the obvious questions that might be asked is "how many generated disruption scenarios are required to represent real situations?". Towards the goal of answering this question, we conducted computational experiments, such that we solved the scenario-based stochastic FDARP under a different number of equally likely scenarios, starting from 50 up to 150, as recommended by experts in the airline. In each experiment, we randomly sampled a number of scenarios by means of a truncated gamma distribution for the length of non-propagated delay, to match the delay data collected from the airline for the mean, second moment, and range. Figure 4.6 represents the computational results of this section, which reveals that the appropriate number of scenarios that can better represent the real situation is 100, as increasing the number of scenarios does not provide significant changes in the obtained solution. This number of scenarios was confirmed by two sources. Firstly, from the experts of the airlines, who recommended that 100 scenarios are more than enough to represent real situations. Secondly, from the literature, this number is confirmed through the work by (Yen and Birge, 2006), who generated disruption scenarios for flight delays, while solving the crew scheduling problem. The previous observations motivate us to set the number of scenarios to be 100 throughout our case study.

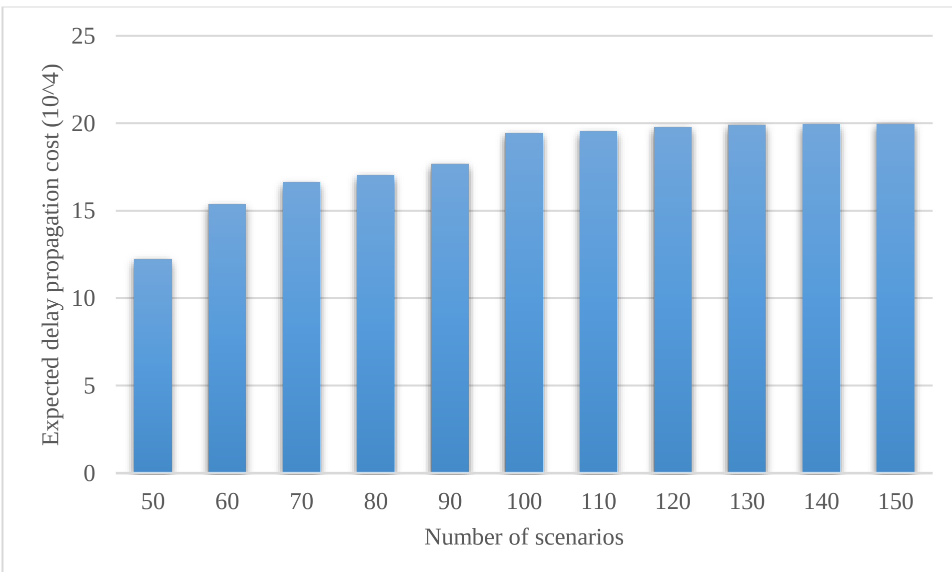


Figure 4.6: Results of scenario-based stochastic FDARP under different scenario numbers

4.5.3 Results of leader-follower Stackelberg game model

In this section, the nested ACO-based algorithm is implemented in order to obtain near optimal solutions for the proposed bi-level optimization model, which is formulated as a leader-follower Stackelberg game. For sake of computational convenience and a meaningful problem context, the nested ACO-based algorithm adopts the following values: α=1, β=2, q_0=0.95, ρ=0.05, Q=0.01, α'=2, β'=2, q_0'=0.85, ρ'=0.05, Q'=0.01, ant size for upper-level ACO=fleet size, and ant size for lower-level ACO=number of flights in which their aircraft are maintained.

Figure 4.7 illustrates the tradeoffs between the expected propagated delay cost of the airline that is handled by the upper-level ACO-based algorithm and the labor cost of the maintenance providers determined by the lower-level ACO-based algorithm. By looking at Figure 4.7, it is obvious that the upper-level ACO-based algorithm reaches the convergence point and gives its best result after 350 iterations. On the other hand, after 450 iterations, the lower-level ACO-based algorithm reaches the convergence point and its best result is achieved. Since these results (at the convergence points) constitute the situation in which all players are unwilling to change their decisions and objective functions, the Stackelberg equilibrium is achieved, with values 16,402.30 for the airline and 232,025.30 for the maintenance providers.

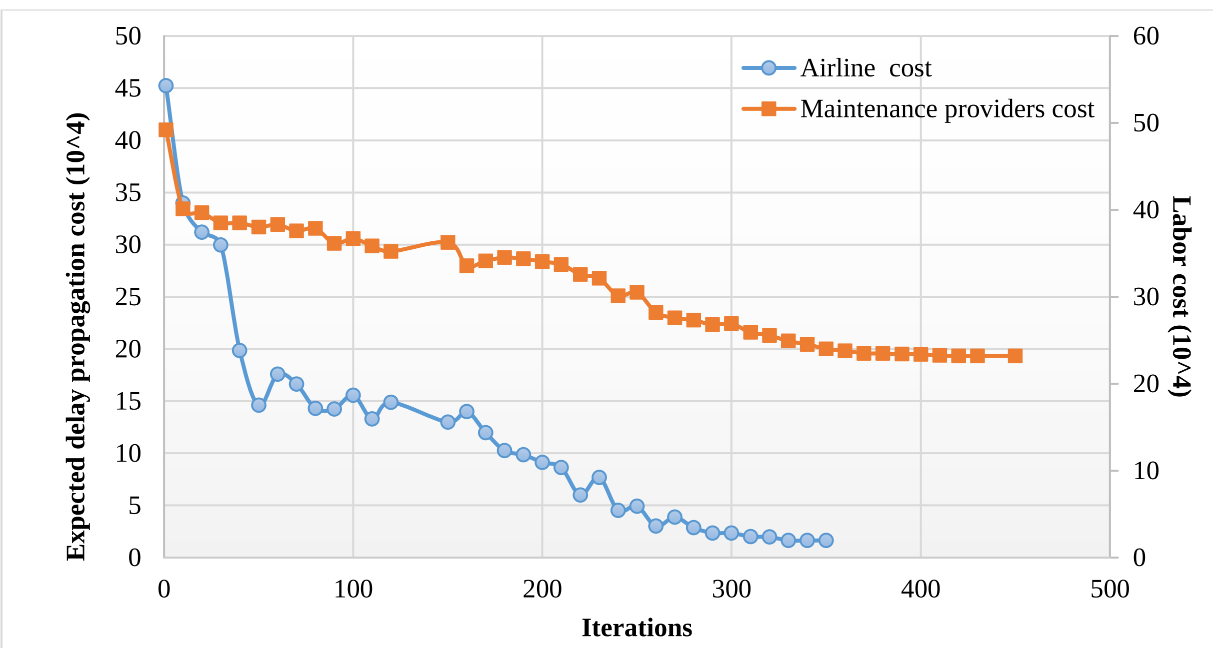


Figure 4.7: Convergence of the nested ACO-based algorithm

4.5.4 Performance of the Stackelberg game model and the existing method

The performance of the developed bi-level optimization model, referred to as the Stackelberg game decision model (LFS) has been provided in the previous section. Presenting the LFS performance is not adequate in demonstrating its advantage over the existing methods in the literature. In this connection, we extend our computational experiments with the aim of comparing the LFS performance with one of the traditional methods that is commonly known in the literature. This method is called the non-joint optimization method (NJOP), which treats each problem of the scenario-based stochastic FDARP and MSP as individual optimization problems and returns separate solutions for each problem. Firstly, NJOP optimizes the scenario-based stochastic FDARP solely on the minimal cost, and then optimizes MSP to minimal cost as well. The results of these computational experiments, while handling the scenario-based stochastic FDARP and MS separately, are presented in Figure 4.8 and Figure 4.9, respectively.

In terms of the expected propagated delay cost of the airline, Figure 4.8 shows that the LFS model outperforms the NJOP model by 15.61% (16,402.30 vs. 19,436.30). Similarly, Figure 4.9 shows further outperformance of the LFS model over the NJOP model by 18.70% (232,025.30 vs. 285,393.90), while handling the labor cost of the maintenance provider. The rationale behind this outperformance lies in the fact that the LFS model optimizes scenario-based stochastic FDARP in conjunction with MSP, which means that the results of one part are sent back to the other part. This results in giving the two parties a chance to continue adjusting their decisions in order to improve the results until reaching equilibrium. In contrast, the NJOP model optimizes both problems separately, which means there is no feedback movement, leading to a loss of opportunity in adjusting the decisions and improving the obtained results.

So, it is clear cut from this section that the proposed LFS model significantly improves the results obtained by the airline and maintenance providers. This echoes the importance of the coordinated decision support system of scenario-based stochastic FDARP and MSP being implemented in reality.

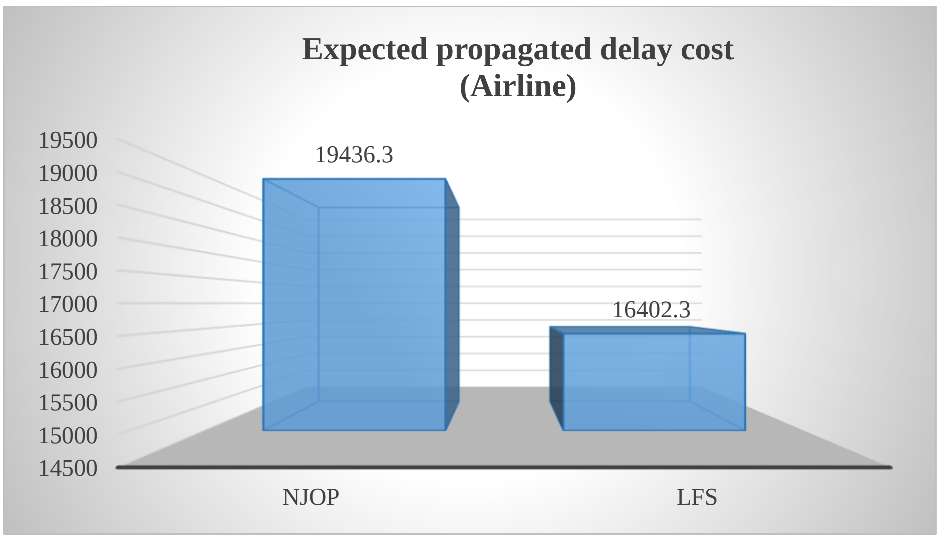


Figure 4.8: Performance comparison for scenario-based stochastic FDARP

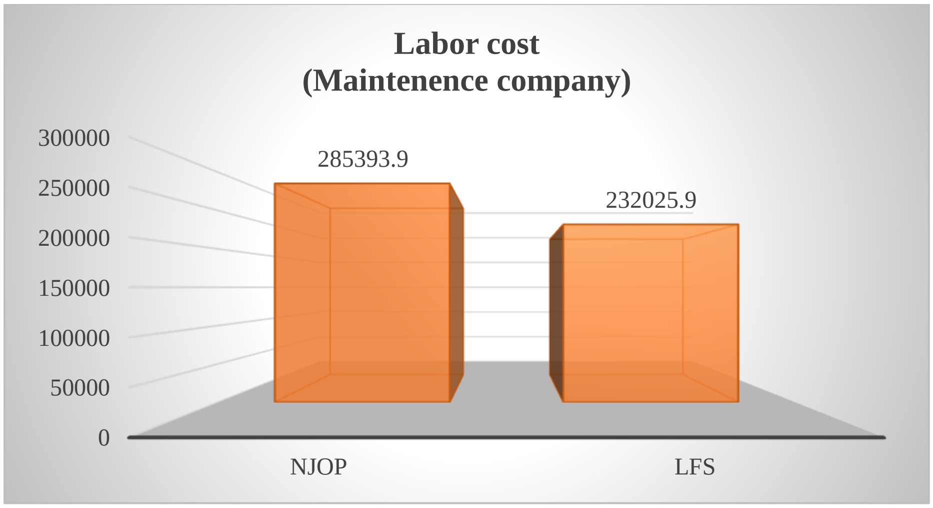


Figure 4.9: Performance comparison for MSP

4.6 Summary

In this chapter, a bi-level optimization model for coordinated decision support system of the scenario-based stochastic FDARP and MSP is proposed by utilizing the Stackelberg game. In this game, the scenario-based stochastic FDARP, which is solved by the airline, acts as a leader with the objective of minimizing the expected propagated delay cost. On the other hand, MSP, which is handled by the maintenance providers, plays the role of the follower that responds rationally to the decisions taken by the leader regarding the real departure time of the airline's aircraft from the maintenance providers' stations. Towards the goal solving the developed bi-level optimization model and achieving the Stackelberg equilibrium, a nested ACO-based algorithm is proposed.

To validate the superiority of the proposed model, a case study of proposed model for a Middle Eastern major airline and four maintenance providers is presented. The results of the case study reveal that both of airline and maintenance providers achieve significant savings in their operational costs, if compared to the results obtained from the traditional non-joint optimization method. This indicates that the proposed model has great potential for implementation in actual practice.

Although this chapter presents a formulation for a unique problem in the literature, there are two main limitations for the proposed model. Firstly, the maintenance providers are serving in a competitive market, in which all the providers are struggling to survive. One of the ways to survive is attracting more maintenance demand from the competitors. To do so, the maintenance providers use the price competition process, which includes cutting down the price of the maintenance service. Observing this action by airlines results in changing their scheduled maintenance visits to target the provider with cheaper maintenance service. This can easily interrupt the routing plan of airlines. So, besides the interdependence among the scenario-based stochastic FDARP of airlines and MSP of maintenance providers, there is another factor that can easily affect the routing plan. Therefore, the price competition among the maintenance providers should be considered besides the interdependence between FDARP of airline and MSP of maintenance providers. Secondly, in this chapter, to capture the non-propagated delays, we propose using scenario-based stochastic framework. Although the successful of this approach to capture the high level of uncertainty of the non-propagated delays, it only focuses on analyzing the historical data and overlooks some other features that affect the delays, including the bad weather, peak seasons, and maintenance station congestion. So, for better forecasting the non-propagated delay, the above described features should be taken into consideration besides the historical data. The aforementioned two limitations are considered in the next chapter.

Chapter 5 - Stackelberg-Nash Game Model for Optimizing the Operational Aircraft Maintenance Routing and Maintenance Staffing with Price Competition Consideration

5.1 Introduction

This chapter is considered the extension of the previous chapter, in which our aim is twofold. Firstly, to accurately forecast the non-propagated delay, such that not only the historical data for the non-propagated delay is considered, but also the other external factors like bad weather, peak seasons, and maintenance station congestion are considered. To consider these enormous data, data analytics is utilized by developing a neural network-based algorithm that couples historical data and the external factors, resulting in a more accurate forecasted NPD. Secondly, to investigate the interdependence between FDARP of airlines and MSP of maintenance providers, while considering the price competition among the maintenance providers. Towards this aim, a Stackelberg-Nash game model (SNGM) is developed, which consists of two sub-games; a leader-follower Stackelberg game (LFSG) to capture the interdependence between FDARP and MSP as in the previous chapter, and a Nash game (NG) to capture the price competition among the maintenance providers. Towards the goal of solving the proposed SNGM and finding out the overall

Nash equilibrium, we develop an iterative game model, in which the nested ACO-based algorithm presented in the previous chapter is coupled with an analytical method. The viability and the potential of the proposed model are demonstrated by using the case study presented in the previous chapter.

The remainder of this chapter is organized as follows. In section 5.2, the model description is presented, whereas the SNGM model formulation is described in section 5.3. To solve the proposed SNGM model, an iterative game algorithm is developed in section 5.4. In section 5.5, we present a neural network-based algorithm to forecast the non-propagated delay. Using the same case study presented in the previous chapter, the potential and feasibility of the proposed model is discussed in section 5.6. Finally, a summary for the chapter is given in section 5.7.

5.2 Model Description

This section mainly describes the proposed SNGM, which consists of two sub-games: the LFSG and the NG. Starting with the LFSG, it reflects the interdependence between the FDARP of airlines and MSP of maintenance providers. Indeed, this game is similar to the game presented in the previous chapter. However, the LFSG in this chapter is different compared to the previous chapter in one aspect, which is the way of capturing the non-propagated delay. As opposed to the scenario-based stochastic formulation in which the non-propagated delays are captured by analyzing the historical data and generating disruption scenarios, in this chapter, the non-propagated delays are forecasted by considering the historical data besides some external factors, including the bad weather, peak seasons, and maintenance station congestion. To consider these enormous data, we utilize data analytics by developing a neural network-based algorithm that combines historical data and the external factors, resulting in an accurate forecasted non-propagated delay. So, the FDARP of airline considers the non-propagated delay by using the forecasted values provided by the neural network-based algorithm.

Let's now move to the second game, called the NG. Recently, the maintenance providers are serving in a tough competitive market because of the economic recession. In order to survive in this competitive market, the maintenance providers are trying to improve their profitability by attracting more demand from airlines. This can be achieved by cutting down the price of maintenance service, which motivates the airlines to increase their demand (i.e. maintenance visits or number of aircraft to be maintained) to maintenance providers with cheaper price. Since the demand of the airline for the maintenance from cheaper providers is changed, the routing plan will be affected because the location of performing the maintenance is changed. So, there is another factor that might interrupt the routing plan of the airlines, called the price competition among the maintenance providers. Therefore, it is imperative to consider the price competition in our model. By looking precisely at this price competition, we can see that this competition is as follows. At the beginning, each maintenance provider cuts the maintenance service price, which price is evaluated by the airlines in terms of the maintenance demand (i.e. number of aircraft to be maintained), resulting in an increase of the demand. Next, the other providers observe this price setting and react towards it by adjusting their prices in order to attract more demand from airlines. If the price setting for each maintenance provider is investigated, we can see that the price of a single provider is not only dependent on the provider's price, but also dependent on the others' prices. This price setting concept leads naturally to formulate the price competition among the maintenance providers as a NG model, with the objective of maximizing the net profit. Indeed, two types of information are required for this game, including the demand and the labor costs, as shown in Figure 5.1. This information is provided by the first game, called LFSG. For the demand, it can be extracted from the routing plan constructed by the FDARP. On the other hand, the labor cost can be determined by the MSP. Using such information helps NG in setting the price for the maintenance service for each maintenance provider. Note that the price by a single provider is not only determined by his pricing decision but also influenced by the others pricing decisions. Consequently, all providers do not have intention to change their prices as their profit is maximized. Therefore, the generated prices form a so-called Nash equilibrium.

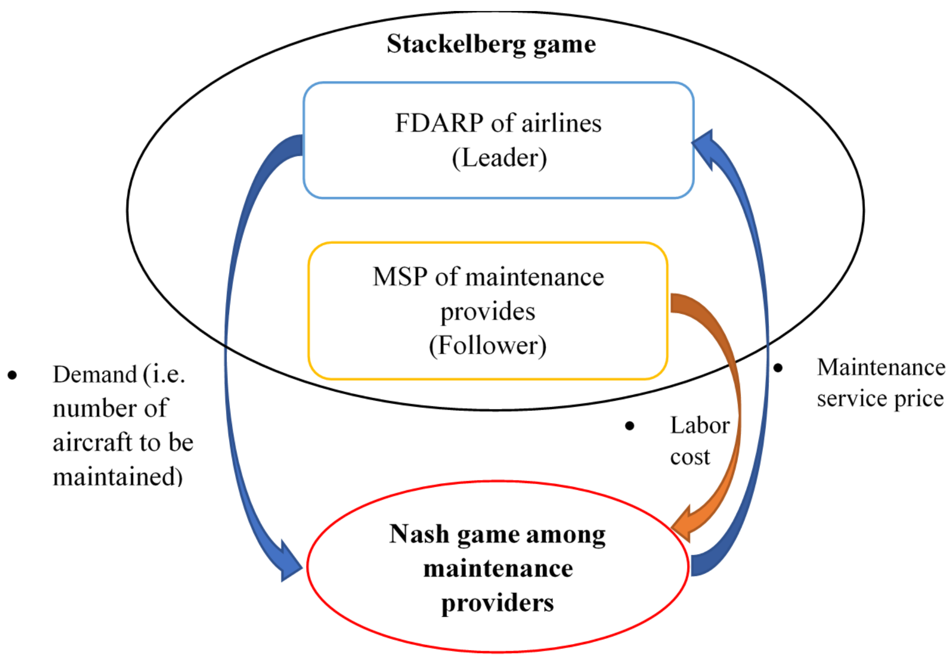


Figure 5.1: The Nash game and its related input and output

The above described sub-games form the SNGM, which is processed as follows. It starts with solving the LFSG, which provides decisions determined by FDARP (i.e. the number of aircraft to be maintained and their corresponding departure/arrival times) and decisions determined by MSP (i.e. the real departure times for the aircraft, team sizes, and the labor cost). Among these decisions, the number of aircraft to be maintained and the labor cost are chosen and sent to the NG. These decisions constitute the input for the NG, which is solved to determine the decisions on the maintenance service prices. These decisions are sent back to the LFSG. If pricing decisions affect the decisions taken by the LFSG, it will be resolved. This process is iterated until reaching the so-called overall Nash equilibrium, in which both LFSG and NG are in equilibrium.

5.3 Stackelberg-Nash Game Model Formulation

This section mainly proposes the formulation of the SNGM, including its two main sub-games: the LFSG and the NG. Before presenting the model formulation, we first define the scope of the model and the notations used throughout the model.

5.3.1 Model scope and notations

The scope of the proposed model can be summarized as follows:

- The model includes a single airline that is served by multiple maintenance providers.
- The airline focuses on solving FDARP using the same scope presented in the previous chapter, regarding the planning horizon, the Type A maintenance check and the limited number of maintenance stations owned by maintenance providers.
- The FDARP of airline considers the expected value of non-propagated delay for each flight leg. In particular, the routing plan is constructed not only by considering

the flight duration of each flight leg, but also by considering the expected non-propagated delay that might happen after each flight leg.

- The MSP of maintenance providers is solved while the workforce capacity is deterministic.
- The maintenance providers offer the maintenance service in a competing market. This competition stems from the fact that Type A maintenance check is the simplest one among the others, as it includes visual inspection of major parts like the aircraft's engine, thus vast majority of providers can provide this check. Therefore, airlines trace the provider with a cheaper service as any provider can perform it. So, due to the competition, each maintenance provider's revenue is not only dependent on his price decision, but also is affected by price decisions taken by other providers.
- Due to the competition, the maintenance service demand for a maintenance provider is not only a function of the price offered by the provider himself, but also a function of prices by all other providers.
- The information exchanges among the maintenance providers are limited; each maintenance provider does not know complete information about the other providers as each provider thinks that it is quite risky to disclose much information. The only action can be taken by each maintenance provider, is reacting towards the price decisions taken by the other providers.

After presenting the scope of the model, we need to summarize the notations used throughout the proposed model. Indeed, in this chapter, we use the same notations presented in the previous chapter, regarding the Stackelberg game, except ignoring any scenarios identified in the decision variables. In addition to the notations used from the previous section, other notations representing the Nash game are used. These notations can be defined as follows:

Airline (Leader of the LFSG)

Sets and indices:

MT Set of maintenance providers, indexed by m or g.

Parameters

$E(NPD_{ik})$: Expected value of the non-propagated delay of flight leg i covered by aircraft k.

Decision variables

$x_{ijkv} \in \{0,1\}$: It takes value of 1 if aircraft k covers two consecutive flight legs i and j, before receiving the maintenance operation number v, and 0 otherwise.

$y_{imkv} \in \{0,1\}$: It takes value of 1 if flight leg i is covered by aircraft k, then the aircraft proceeds to maintenance provider m to receive the maintenance operation number v, and 0 otherwise.

$z_{mjkv} \in \{0,1\}$: It takes value of 1 if aircraft k leaves maintenance provider m to cover flight leg j, after receiving the maintenance operation number v, and 0 otherwise.

$RTAM_{kv}^{*} > 0$: The ready time when aircraft k completes receiving the maintenance operation number v and able to cover the next scheduled flight legs.

Maintenance providers (Follower of LFSG + Forming the NG)

Parameters

De_m: Demand volume of maintenance service for maintenance provider m.

θ_m: A positive constant that reflects the potential size of maintenance service that can be offered by maintenance provider m.

ϑ_m: Sensitivity of maintenance service demand for maintenance provider m to its maintenance service price.

δ_{mg}: Sensitivity of maintenance service demand for maintenance provider m to the maintenance service price set by competitor g.

TA_m: Total number of aircraft served by maintenance provider m.

NP_m: Profit of maintenance provider m.

Decision variables for maintenance providers

$wf_{fsm} \in \{w_{sm}^{l}, \dots, w_{sm}^{u}\}$: Number of workers (team size) assigned by maintenance provider m to maintain aircraft that covers flight f during shift s.

$RTAM_{fm} > 0$: Actual ready time when the aircraft that covers flight f completes its maintenance by maintenance provider m, known as the real departure time for the aircraft that covers flight f.

$CC_{fsm} \in \{0,1\}$: =1 if the aircraft that covers flight f received the maintenance by maintenance provider m during shift s and 0 otherwise.

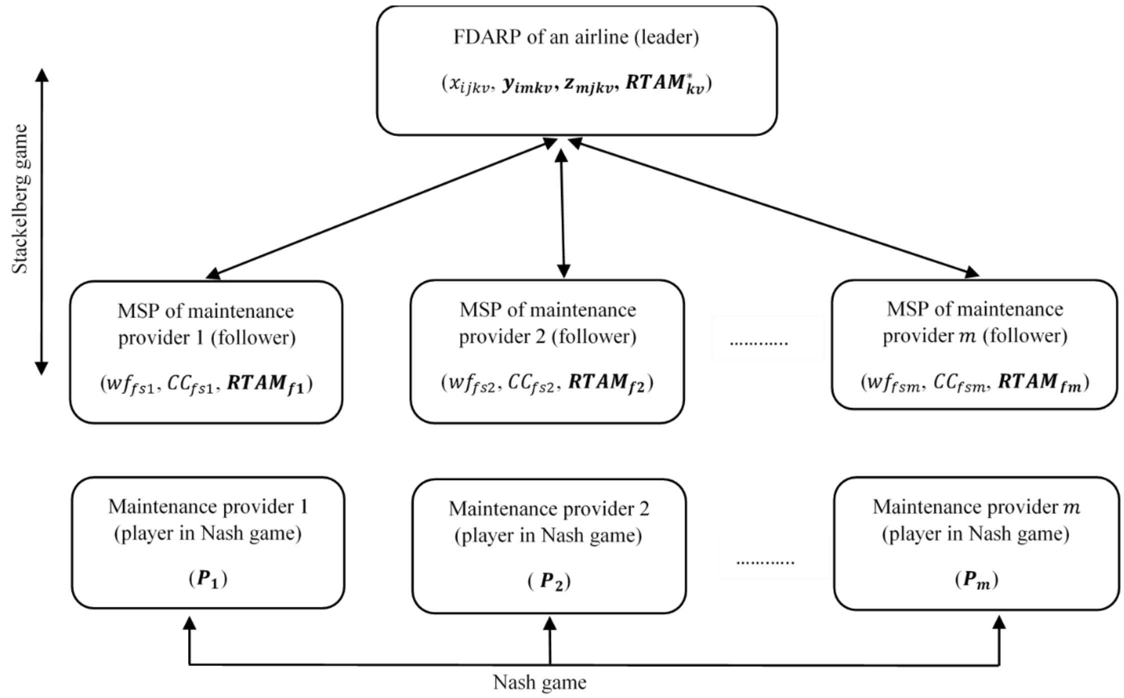
$P_m > 0$: Price of the maintenance service offered by maintenance provider m.

5.3.2 Framework of the SNGM

In Figure 5.2, the framework of the SNGM is elaborated. Indeed, this model consists of two sub-games: a vertical LFSG and a horizontal NG.

For vertical LFSG, it represents the inherent interdependence between an airline and multiple maintenance providers. In this game, the airline acts as a leader by solving FDARP in order to determine the routing plan decisions (referred to as y_{imkv}, z_{mjkv}, and $RTAM^*_{kv}$). Actually, the first two decisions indicate the demand of the airline as y_{imkv} and z_{mjkv} represents the number of aircraft to be maintained, whereas the last decision reflect the departure times of the aircraft. The airline sends these decisions to the maintenance providers, who in turn act as follower and solve MSP to determine the staffing plan decisions, (referred to as $RTAM_{fm}$). This decision represents the time when the aircraft complete the maintenance operation and ready to leave the maintenance station. known as the real departure time (i.e. ready time) for aircraft. The decisions by the airline and maintenance providers are kept exchanged until finding out the Stackelberg equilibrium, in which all players do not have intention to change their decisions because any deviation might bring negative induce to their objective functions.

The second game represented by the SNGM is called NG, in which the competition among maintenance providers is captured. This game is started by receiving two decisions from LFSG. Firstly, the routing decision y_{imkv}, which indicates the demand for the maintenance providers. Secondly, the staffing plan decision wf_{fsm}, which used to calculate the labor cost by help of C_{wfsm}. These decisions are used in the process of calculating the maintenance service price decision for each provider (referred to as P_m). It is noteworthy that these prices are calculated not only by considering the self-pricing decisions, but also by considering the others' prices decisions. Therefore, all providers do not have intention to change their prices, resulting in a generation of the Nash Equilibrium. The pricing decisions by NG are sent back to the LFSG. If these prices cause a change on the Stackelberg decisions, the LFSG will be resolved to generate new routing and staffing decisions. Next, the new decisions are sent to the NG and so on. This process is iterated until reaching a stable situation, when all the players are unwilling to change their decisions, as any deviations cannot improve their own benefit. This stable situation is called the overall Nash equilibrium, in which the LFSG and the NG are in equilibrium.



Note: (1) The variables in the brackets are the decision variables of each player; (2) The decision variables in bold are those values can affect the others benefits.

Figure 5.2: Framework of the Stackelberg-Nash game

5.3.3 Formulation of the LFSG

In this section, we present the LFSG, which is modeled as a bi-level optimization model. The upper-level part of the bi-level optimization is represented by the FDARP of airline, whereas the lower-level part is captured by the MSP. Indeed, this model is similar to that one presented in the previous chapter, except adding the maintenance service cost to the objective function of the FDARP and neglecting the scenario generated in the FDARP. Based on the predefined notations in this chapter and the previous one, the LFSG as a bi-level optimization model can be formulated as follows:

$$\min \sum_{v=1,\dots,V} C_{pD}\left(\sum_{k\in K}\sum_{i\in I}\sum_{j\in I} PD_{ijkv}\, x_{ijkv}\right)+\sum_{m\in MT} P_m\left(\sum_{i\in I}\sum_{k\in k}\sum_{v=1,\dots,V} y_{imkv}\right) \tag{5.1}$$

s.t. $PD_{ijkv} = PD_{ikv} + (E(NPD_{ik}) - (DT_j - AT_i - TRT))^+ \quad \forall\, i,j \in I, \forall\, k \in K, \forall\, v = 1,\dots,V$ (5.2)

Constraints in Eqs. (4.3) – (4.18) (5.3)

$x_{ijkv} \in \{0,1\} \quad \forall\, i,j \in I, \forall\, k \in K, \forall\, v = 1,\dots,V$ (5.4)

$y_{imkv} \in \{0,1\} \quad \forall\, i \in I, \forall\, m \in MT, \forall\, k \in K, \forall\, v = 1,\dots,V$ (5.5)

$z_{mjkv} \in \{0,1\} \quad \forall\, m \in MT, \forall\, j \in I, \forall\, k \in K, \forall\, v = 1,\dots,V$ (5.6)

$RTAM_{kv} > 0 \quad \forall k \in K, \forall\, v = 1,\dots,V$ (5.7)

where given decision variables (y_{imkv}, z_{mjkv}, and $RTAM^*_{kv}$) are used for solving:

$$\min \sum_{m\in MT}\sum_{s\in S}\sum_{f\in F} C_{wfsm}\, wf_{fsm} CC_{fsm} \tag{5.8}$$

s.t. Constraints in Eqs. (4.24) – (4.28) (5.9)

$wf_{fsm} \in \{w^l_{sm},\dots,w^u_{sm}\} \quad \forall f \in F, \forall s \in S, \forall m \in MT$ (5.10)

$RTAM_{fm} > 0 \quad \forall f \in F, \forall m \in MT$ (5.11)

$CC_{fsm} \in \{0,1\} \quad \forall f \in F, \forall s \in S, \forall m \in MT$ (5.12)

The upper-level part of the bi-level optimization model is represented by the FDARP in Eqs. (5.1) - (5.7), whereas the lower-level part of the bi-level optimization model is represented by the MSP in Eqs. (5.8) - (5.12). As mentioned earlier, the bi-level optimization model presented here is similar to the bi-level optimization model that is presented in the previous chapter, except adding one term in the objective function of the upper-level part. This term is the maintenance service cost. It is important to mention here that the maintenance cost is calculated according to the maintenance service prices determined by the NG.

5.3.4 Formulation of the NG

In this section, the NG is proposed, which captured the way that the maintenance providers compete among themselves while setting the price of the maintenance service. The NG can be formulated as follows:

Maximize (for $\forall m \in MT$):

$$NP_m = De_m P_m - \left(\frac{\sum_{s\in S}\sum_{f\in F} C_{wfsm} wf_{fsm} CC_{fsm}}{TA_m}\right) De_m \tag{5.13}$$

where $$De_m = \theta_m - \vartheta_m P_m + \sum_{g\in MT} \delta_{mg} P_g \tag{5.14}$$

$$TA_m = \sum_{i\in I}\sum_{k\in k}\sum_{v=1,\dots,\Psi} y_{imkv} \tag{5.15}$$

Eq. (5.13) express the profit of the maintenance provider m. In particular, the profit can be calculated based on the revenue as shown by the first term and the total labor cost as represented in the second term. Note that the total labor cost is determined by multiplying the demand (De_m) by the average labor cost incurred for each aircraft $((\sum_{s\in S}\sum_{f\in F} C_{wfsm} wf_{fsm} CC_{fsm})/TA_m)$. As mentioned earlier, the average labor cost is determined based on two sources of information. Firstly, the decision variables wf_{fsm} and CC_{fsm} that are received from the follower of the LFSG. Secondly, the decision variable y_{imkv} that is received from the leader of the LFSG. This decision variable helps in specifying the TA_m, as shown in Eq. (5.15).

Since maintenance providers compete among themselves while setting the maintenance service prices, each maintenance provider's demand should be formulated based on its own price and the other observed prices. Therefore, the demand is formulated in consistent with this observation, as shown in Eq. (5.14), such that the demand for maintenance provider m is not only function of its own price P_m, but also function of other competitor's prices P_g. It should be noted that ϑ_m and δ_{mg} are given while considering the demand properties: $\frac{\partial De_m}{\partial P_m} < 0, \frac{\partial De_m}{\partial P_g} > 0, m, g \in MT$ that can go back to Samuelson (1947).

5.4 Solution Algorithm for Overall Nash Equilibrium

In order to get the overall Nash equilibrium, it is necessary to get two kinds of equilibrium: (1) the Stackelberg equilibrium for the LFSG, and (2) the Nash equilibrium for the NG, which are shown in the next two sub-sections. Next, we propose an algorithm in order to find the overall Nash equilibrium.

5.4.1 Obtaining the Stackelberg equilibrium

To get the Stackelberg equilibrium, it is necessary to solve the bi-level optimization model. Since the bi-level optimization model presented in this chapter is similar to that one presented in the previous chapter except the maintenance service cost included the objective function, we can use the nested ACO-based algorithm proposed in the previous chapter to obtain the solve the bi-level optimization model with two modifications. These modifications are: (1) adding one more step in the upper-level ACO-based algorithm regarding the selection of the appropriate maintenance station and, (2) neglect all the generated scenarios as we forecast the non-propagated delay using a neural network-based algorithm.

The added step is called the visiting maintenance providers. Indeed, this step is conducted by the help of the ants by scouting throughout the network and select the appropriate maintenance provider using a so-called state transition rule. In other words, suppose that an ant covers a flight leg represented by node i, and looks for covering next flight leg represented by node j such that its destination has a maintenance provider. To select the next flight leg, we adopt the following state transition rule:

$$j = \begin{cases} \arg\_max_{j \in NVM_i^k} \left\{ \left[\tau_{jm}\right]^{\alpha} \left[\eta_{jm}\right]^{\beta} \right\} & if\ q \leq q_0 \\ J & if\ q > q_0 \end{cases} \quad (5.16)$$

where NVM_i^k denotes the group of possible flight legs that ant k can select after covering flight leg i, such that these flight legs offer maintenance providers in their destination airports. The terms τ_{jm} and η_{jm} represent the pheromone trail and the heuristic function of the maintenance arc $maint(j, m)$, respectively. It is important to mention here that η_{jm} is equal to $1/P_m$. The relative importance of τ_{jm} and η_{jm} is denoted by parameters α and β, respectively. q_0 is the exploration threshold parameter $(0 \leq q_0 \leq 1)$ and q represents random number that is generated according to the uniform distribution $[0 \sim 1]$. Ideally, the value q guides the ant to select the next flight leg j. In case of $q \leq q_0$, the ant selects flight leg j which its arc $maint(j, m)$ carries the greatest τ_{jm} and η_{jm} . Conversely, in the case of $q > q_0$, the flight leg j is selected using the probability rule below:

$$P_{jm}^k = \frac{\left[\tau_{jm}\right]^{\alpha} \left[\eta_{jm}\right]^{\beta}}{\sum_{j \in NVM_i^k} \left[\tau_{jm}\right]^{\alpha} \left[\eta_{jm}\right]^{\beta}} \qquad if\ j \in NVM_i^k \quad (5.17)$$

While solving the bi-level optimization model, we need to specify how to calculate the Stackelberg equilibrium. To simplify the explanation of Stackelberg equilibrium, we let X_0 represents the decision variables (x_{ijkv}, y_{imkv}, z_{mjkv} and $RTAM^*_{kv}$) taken by FDARP of airline, whereas $X_{m,Stack}$ is designed to denote the decision variables (wf_{fsm}, CC_{fsm}, $RTAM_{fm}$) taken by MSP of maintenance providers. Based on the previous definitions, the response functions of FDARP and MSP can be defined in Eqs. (5.18), and (5.19), respectively. It means that the decision, X_0, of FDARP is a function, $r_0(.)$, of the variable $X_{m,Stack}$ taken by MSP. Similarly, in Eq. (5.19), it indicates that the decision taken by MSP is a function of decision taken by FDARP.

$$X_0 = r_0(X_{m,Stack}) \tag{5.18}$$

$$X_{m,stack} = r_m(X_0) \tag{5.19}$$

For achieving the Stackelberg equilibrium, the upper-level ACO based algorithm and the lower-level ACO-based algorithm are used in a dynamic reaction process with multiple iterative stages. Suppose at given iterative stage t, with the decision $X^t_{m,Stack}$ taken by MSP, and the decision X^t_0 taken by FDARP, the FDARP and MSP would make a response as shown in Eqs. (5.20) and (5.21), to obtain their decisions at iterative stage t+1.

$$X^{t+1}_0 = r_0(X^t_{m,Stack}) \tag{5.20}$$

$$X^{t+1}_{m,stack} = r_m(\,X^t_0) \tag{5.21}$$

The Stackelberg equilibrium can be achieved if the following conditions are existed and satisfied (Liu, 1998, Yu and Huang, 2010):

$$\left\| X_0^{t+1} - r_0(X_{m,Stack}^t) \right\| == 0 \tag{5.22}$$

$$\left\| X_{m,stack}^{t+1} - r_m(\, X_0^t) \right\| == 0 \tag{5.23}$$

This means that FDARP and MSP are not willing to change their decisions as any change might induce negative impact for their objective functions. Therefore, the bi-level ACO-based algorithm is terminated.

5.4.2 Obtaining the Nash equilibrium

Realizing the Nash equilibrium necessitates solving the NG model. By looking at the NG model, we can observe that the net profit function is a continuous and differentiable function. Therefore, this NG model can be solved by adoption of the standard optimization approaches like partial differentiation with respect to prices (Hsu et al., 2010, Yu and Huang, 2010). It is worth to note here that the discrete decision variables C_{wfsm}, and y_{imkv} are determined by the LFSG, and their values are used in the NG. So, the profit function still continuous and differentiable. Based on the previous observation, we can get the optimal decision on P_m for all the maintenance providers by using the following equation:

$$\frac{\partial NP_m}{\partial P_m} = 0 \qquad \forall\, m \in MT \tag{5.24}$$

To simplify the calculation of Nash equilibrium, we design $X_{m,Nash}$ to represent the decision variable P_m taken by maintenance provider m. For any maintenance provider m, the decision variables of all other maintenance providers can be expressed as $X_{-m,Nash}$. Based on the previous definitions, the response functions of maintenance provider m can be defined as:

$$X_{m,Nash} = r_m(X_{-m,Nash}) \tag{5.25}$$

It means that the decision, X_m, taken by maintenance provider m is a function of the variable $X_{-m,Nash}$ taken by all other providers. For achieving the Nash equilibrium, the maintenance providers normally are behaving in a dynamic way with multiple iterative stages. Suppose at given iterative stage t, with the decision $X^t_{-m,Nah}$ taken by all other maintenance providers, the maintenance provider m would make a response as shown in Eqs. (5.25) to obtain its decisions at iterative stage t+1.

$$X^{t+1}_{m,Nash} = r_m(X^t_{-m,Stack}) \tag{5.26}$$

The Nash equilibrium can be achieved if the following conditions are satisfied (Liu, 1998, Yu and Huang, 2010):

$$\sum_{m \in MT} \left\| X_0^{t+1} - r_0(X^t_{m,Stack}) \right\| == 0 \tag{5.27}$$

This means that all the maintenance providers are not willing to change their pricing decisions as any change might cause losses to their profits.

5.4.3 Obtaining the overall Nash equilibrium

Realizing the overall Nash equilibrium entails an algorithm that can achieve both the Stackelberg equilibrium and Nash equilibrium simultaneously. For this purpose, we propose an iterative game algorithm that couples the bi-level ACO-based algorithm and the analytical method described in the previous sections. The detailed procedures of the iterative game algorithm are as follows:

Step 1: Initialize the parameters values of the bi-level ACO-based algorithm (i.e. $\alpha, \beta, q_0, \rho, Q, \alpha', \beta', q_0', \rho', Q'$, and the number of ants). Then, set a value for the maximum number of iterative stages $t \in T$.

Step 2: Initialize the number of iterative stages $t = 1$.

Step 3: Determine the routing plan decisions by applying Steps 1.7- 1.9 of the nested ACO-based algorithm described in the previous chapter.

Step 4: For the existing iterative stage t, store the routing plan decisions in X_0^t. Next, calculate the solution of this stage and update the best solution found so far.

Step 5: Determine the staffing plan decisions by applying Step 2.6 of the nested ACO-based algorithm described in the previous chapter.

Step 6: For the existing iterative stage t, store the staffing plan decisions of maintenance provider m in $X_{m,Stack}^t$.

Step 7: For the existing iterative stage t, calculate the solution, update the best solution found so far and go to Step 3.

Step 8: Determine the maintenance service price P_m for each maintenance provider by applying Eqs. (5.24)

Step 9: By using Eqs. (5.22), (5.23), and (5.27), check whether the Stackelberg equilibrium and the Nash equilibrium are achieved or not. If both equilibriums are achieved, go to Step 10, else, increment the iterative stage and go to Step 3.

Step 10: Since the Stackelberg equilibrium and the Nash equilibrium are achieved, the overall Nash equilibrium is obtained. Then, terminate the algorithm.

Figure 5.3 elaborates the flowchart of the iterative game algorithms. The upper-part of the figure shows the bi-level ACO-based algorithm, which used to handle the LFSG, whereas the lower-part of the figure illustrates the analytical method used to solve the NG. For the sake of computational convenience, we set the maximum number of iterative stages to be 500 stages.

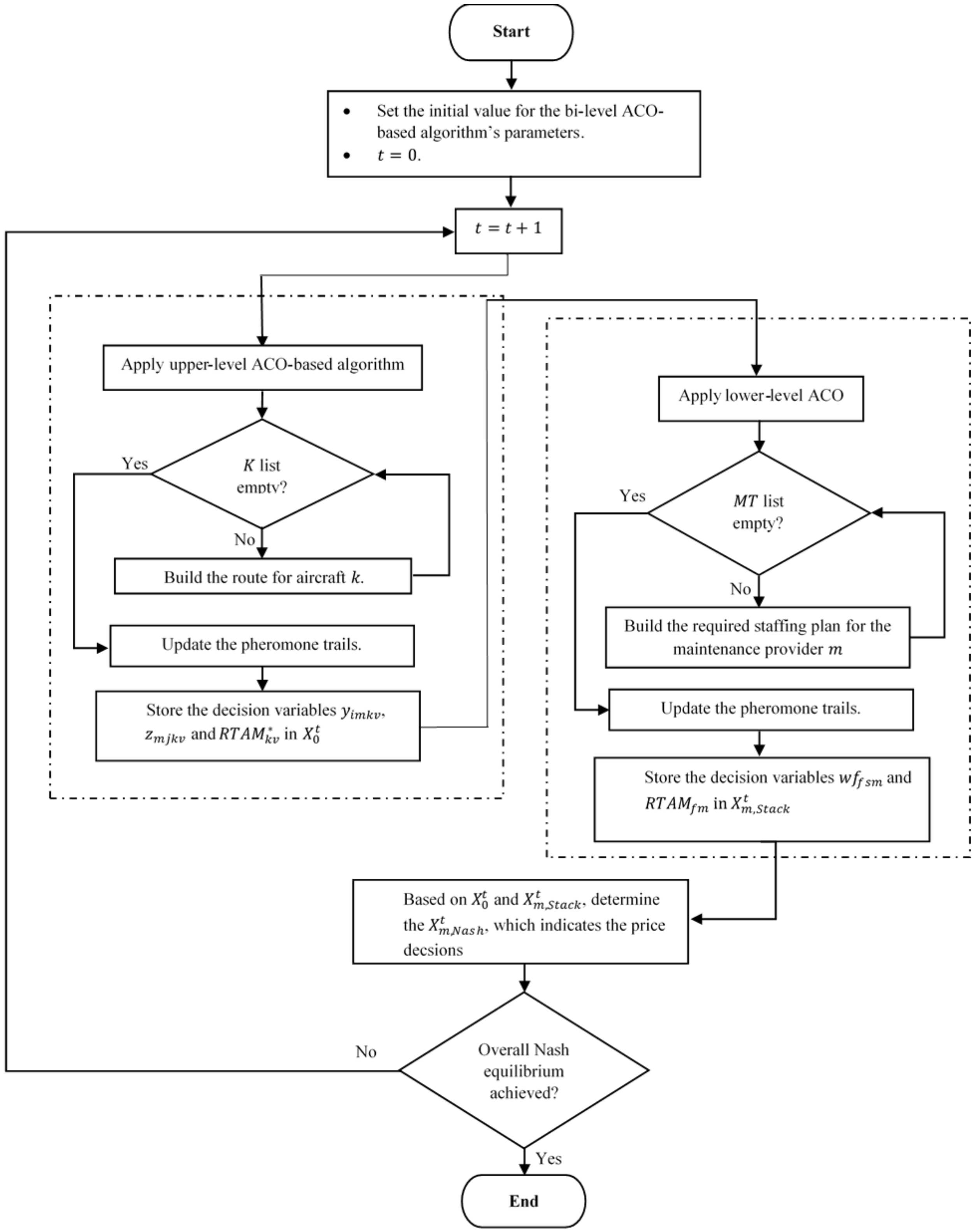


Figure 5.3: Flowchart of the iterative game algorithm

5.5 Data Analytics for Non-propagated Delay Forecasting

Using the iterative game algorithm to solve the proposed SNGM is not enough to conduct the experiments of model presented in this chapter, as it misses the way to calculate the non-propagated delay (NPD). To forecast the NDP, in contrast to the expected value approach that focused on the historical data, we adopt data analytics technique that is able to consider massive information besides the historical data. This adoption includes developing a neural network-based algorithm, as being a promising tool to capture the nonlinear relationship among various factors that affect the NPD. The main steps of this algorithm can be summarized as follows:

Step a: Data collection. The data is collected from a major airline in the Middle East, including some features such as flight number, departure airport, arrival airport, arrival time, departure time, flight duration, the NPD for each flight, and others.

Step b: Data preprocessing. For the collected data, any flight is considered as a delayed flight even if its related NPD time is less than 15 minutes, as any NPD time may easily cause a propagated delay in practice. Meanwhile, any NPD longer than 170 minutes is discarded, as it reflects a severe disruption, which is out of the scope of our research study.

Step c: Define the input sets. Indeed, these sets include historical information and other factors that affect the NPD. These sets can be summarized as follows:

I. Set 1: flight number, departure airport, departure time, arrival airport, arrival time, visited maintenance station, day of operation, and flight duration.

II. Set 2: bad weather indicator. It is known that the NPD frequently occurs during the bad weather. Since it is difficult to predict the time of bad weather, a three-point scale indicator is proposed, in which the values of 1, 2, and 3 indicating less chance, medium chance, and high chance of bad weather occurrence, respectively.

III. Set 3: maintenance station congestion indicator. The NPD can be caused due to delay in the maintenance stations in cases of congestion. To capture this situation, we use four-point scale indicator, with values of 1, 2, 3, and 4, which indicating below 30% station utilization, 30%-60% station utilization, 60%-80% station utilization, and over 80% station utilization, respectively.

IV. Set 4: season indicator. It is known that the NPD frequently occurs during the season, like Christmas and summer vacations. In this sense, three-point scale indicator is used, in which value of 1, 2, and 3, indicating normal day, one week before or after the season, and the season, respectively.

Step d: Design the structure of the neural network. We adopt a multilayer feed-forward neural network, as being a structure commonly adopted. This network consists of input layer, hidden layer, and output layer. For activation function, we use the sigmoid function, as being an efficient function to capture the non-linear relationship between different factors.

Step e: Train the neural network. For this purpose, we use the supervised learning method, in which 70% of data is used for training, and the rest of the data is used for validation purpose.

5.6 Case Study

5.6.1 Problem context

After proposing the SNGM that captures two issues; the coordination between the FDARP of airline and MSP of maintenance providers, and the competition among the maintenance providers, it is necessary to demonstrate the effectiveness of the proposed model as a decision tool for airline and maintenance providers. For this purpose, we use the same case study presented in the previous chapter in order to validate our proposed model. The proposed algorithm and model were coded in MATLAB R2014a, and tested on an Intel i7 CPU processor with 2.50 GHz CPU clock speed and 8 GB RAM laptop, running Windows 10.

5.6.2 Non-propagated delay forecasting

Using the data presented in the previous section is not enough to conduct the experiments of this study as it misses the NPD. To get the NDP, we apply the proposed neural network-based algorithm. For this purpose, we collect the information for all flights recorded by the airline from January 2017 to December 2017, including flight number, departure airport, departure time, arrival airport, arrival time, visited maintenance station, day of operation, and flight duration. The data contains a total of 292,000 flights flown by 12 fleets. After analyzing these data, we picked the top fleet with the longest average propagated delay, to test the capability of the proposed model to minimize the propagated delay (PD), and to test the potential of the proposed neural network-based algorithm to forecast accurate NPD. The features of the picked fleet are summarized in Table 5.1.

Table 5.1: Features of the selected fleet

Fleet	**Total flights**	**Delayed flights**		**NPD (minutes)**		**PD (minutes)**	
		No.	%	Total	Average	Total	Average
A300	29,440	6,597	22.41	333,849	11.34	287,628	9.77

5.6.3 Data analytics for predicting the demand-price function for maintenance providers

Conducting the experiments of this study necessities predicting the demand-price function for maintenance providers. For this purpose, data analytics in the form of regression is adopted, as being one of the most efficient tools to capture the relationship between a response variable and one or multiple predictors. Since the demand for a maintenance provider is a function of the price offered by the provider himself and the prices offered by all other provider, the demand is a function of multiple predictors. So, it is reasonable to adopt the multiple linear regression algorithm in this study. It should be noted that this regression algorithm was conducted based on real collected data from the maintenance providers for the period from January 2017 to December 2017.

The multiple linear regression algorithm is used to obtain the demand-price function for each maintenance provider, as shown in Table 5.2. To assess the quality of the obtained relationship, two indicators are used. First indicator is the R-squared, which indicates how well the obtained model fits the collected data. In a close look at results in Table 5.2, we can notice that R-squared indicator for all functions is larger than 90%, indicating that the regression model fits the collected data extremely well. The second indicator is the p-value, which indicates the relationship between the response variable and the predictors. If the obtained p-value is larger than the selected significance level, there is no significant relationship between the response variable and the predictors. By checking the p-values presented in Table 5.2, we can notice a significant relationship between the demand and the prices as the p-values are smaller than the significance level, which is 5% in this analysis.

Table 5.2: Regression analysis between the demand and its related prices

Maintenance provider	**Regression fitted line** $\boldsymbol{De_m = \theta_m - \vartheta_m P_m + \sum_{g \in MT} \delta_{mg} P_g}$	**R-squared**	**Predictor variable**	$\boldsymbol{p}$**-value**
1	De_1= 35 - 0.00428 P_1+ 0.00117 P_2+ 0.00150 P_3 + 0.000842 P_4	98.7%	θ_1	0.001
			P_1	0.000
			P_2	0.000
			P_3	0.000
			P_4	0.000
2	De_2= 30 - 0.00296 P_2+ 0.00105 P_1+ 0.00097 P_3 +0.000604 P_4	94.5%	θ_2	0.000
			P_2	0.000
			P_1	0.000
			P_3	0.000
			P_4	0.000
3	De_3= 32 - 0.00348 P_3+ 0.00121 P_1+ 0.00120 P_2 +0.000996 P_4	95.4%	θ_3	0.000
			P_3	0.000
			P_1	0.000
				0.000

			P_2 P_4	0.000
4	De_4= 27 - 0.00235 P_4+ 0.000900 P_1+ 0.00049 P_2 +0.00059 P_3	91.1%	θ_4 P_4 P_1 P_2 P_3	0.000 0.000 0.000 0.000 0.000

5.6.4 Results of the Stackelberg-Nash model

In this section, we report the result obtained by solving the Stackelberg-Nash model using the iterative game algorithm. To use the iterative game algorithm, we need to set values for its parameters. For sake of computational convenience, the iterative game algorithm adopts the following values; α=1, β=2, q_0=0.95, ρ=0.05, Q=0.01, α'=2, β'=2, q'_0=0.85, ρ'=0.05, Q'=0.01, ant size for upper-level ACO=fleet size, and ant size for lower-level ACO=number of flights in which their aircraft are maintained.

Implementing the iterative game algorithm provides the results shown in Figures 5.4 and 5.5. Figure 5.4 illustrates the results of the LFSG that acts as a coordinated decision support system between the FDARP of airline and the MSP of maintenance providers. By looking at the Figure 5.4, we can see that after 450 iterative stages, the algorithm reaches the convergence point, meaning that all players are unwilling to change their decisions, resulting in an overall Nash equilibrium values of 318,000 for the airline, and 207,520 for the maintenance providers. Figure 5.5, on the other hand, shows the results of the NG among the maintenance providers, including the prices and the net profits achieved at the overall Nash equilibrium.

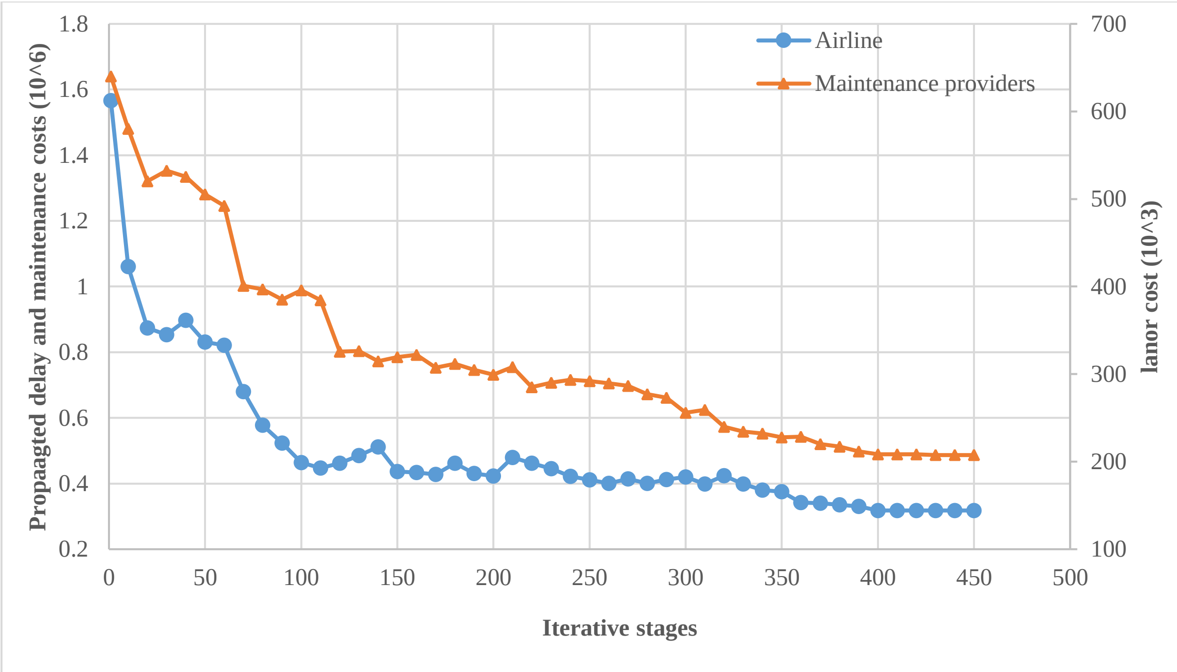


Figure 5.4: Convergence of the iterative game algorithm

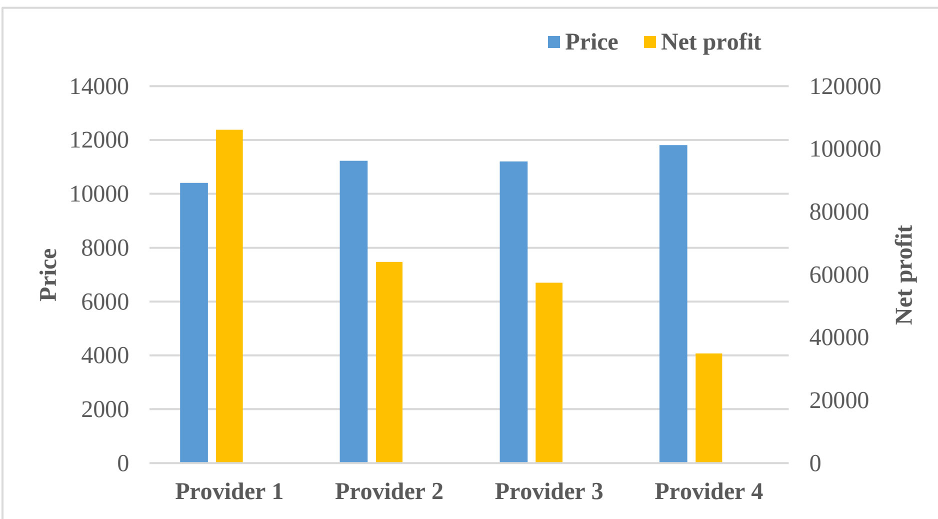


Figure 5.5: Price and net profit for maintenance providers

As pointed out earlier, the proposed model consists of two sub-games: the LFSG and the NG. Presenting the performance of the proposed model in an overall raises some questions; what the role and importance of each game of the model are, what is the reason behind the model performance, is it due to considering the coordination using the LFSG or due to considering the competition through the NG. Towards the goal of answering these questions, our experiments are extended to compare between different setting for the proposed model, as shown in the following sections.

5.6.5 Importance of the NG

To discover the importance of the NG that captures the competition among the maintenance providers, we need to compare between two situations; considering the competition and neglecting the competition. The first situation can be achieved by the proposed model in the previous section, in which both the coordination through the LFSG and the competition through the NG are considered. The second situation, on the other hand, can be represented by a model that only considers the coordination through the LFSG, so that the competition is neglected. The second model can be captured by Eqs. (5.1) -(5.12) and be solved by the bi-level ACO-based algorithm. This bi-level ACO-based algorithm can be implemented by applying all steps of the iterative game algorithm, while neglecting step 8, the Nash equilibrium consideration in step 9, and any prices set by the NG throughout the whole algorithm.

The results of the bi-level ACO-based algorithm are shown in Figure 5.6, in which the algorithm converges after 500 iterative stages, and returns the Stackelberg equilibrium values of 380,000 for the airline, and 207,250 for the maintenance providers.

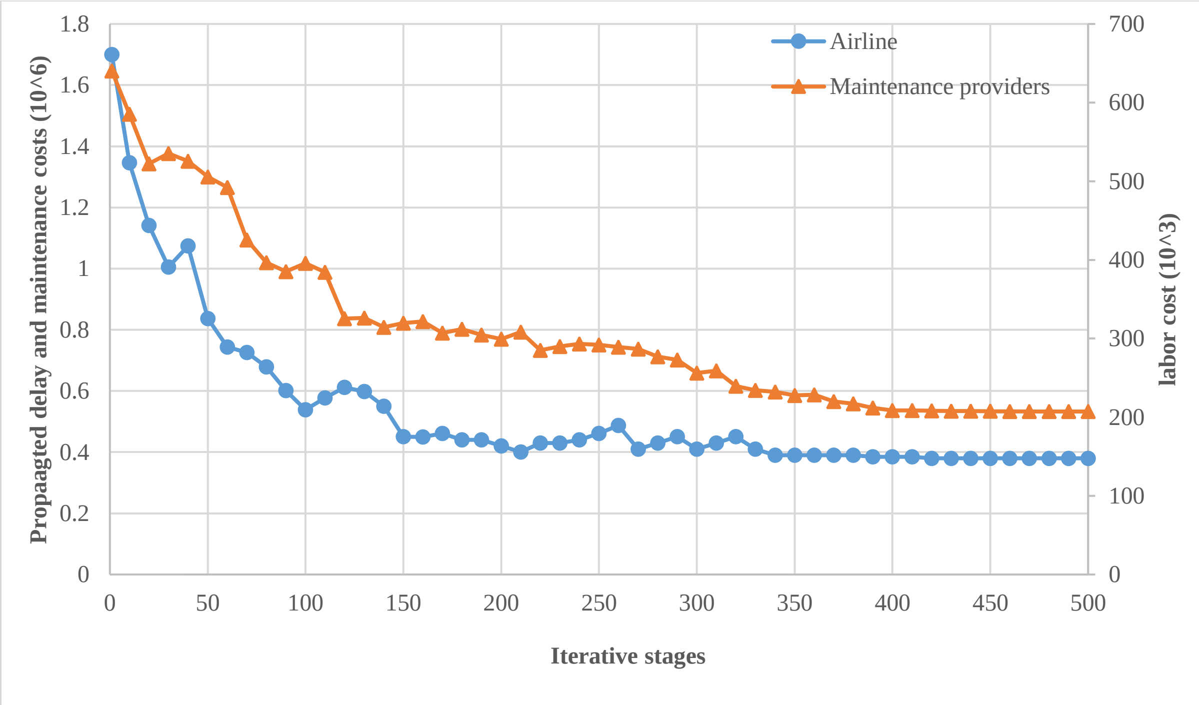


Figure 5.6: Convergence of the bi-level ACO-based algorithm

The performance of the two models in handling the airline and maintenance providers costs are given in Table 5.3. By looking at the results, we can see that the first model outperforms the second one by about 16.32%, while handling the airline costs. The rationale behind this outperformance lies in the fact that considering the competition among the maintenance providers, which includes cutting the maintenance service prices, provides airline the opportunity to trace providers with cheaper prices. This results in a reduction in the cost paid by airline, as in the first model. In contrast to the first model, the second model neglects the competition, so that the airline loses the opportunity of tracing cheaper prices due to competition, resulting in a higher maintenance cost. For labor cost by maintenance providers, it is mainly affected by the coordination between the airline and the maintenance providers, which includes adjusting the staffing plan decisions till reach the Stackelberg equilibrium. Since the two models consider the coordination through the LFSG, there is no expected change in the labor cost.

Table 5.3: The performance of the first and second models while handling the airline and maintenance providers costs

Costs	First model (Coordination + Competition)	Second model (Only Coordination)	Outperformance (%)
Propagated delay and maintenance costs by airline	318,000	380,000	16.32
Labor cost by maintenance providers	207,500	207,500	0

Table 5.4 reports the results of the two models, while handling the net profit of the maintenance providers. We can see from Table 5.4 that three out of four providers achieve better profit while using the first model instead of using the second model. The reason for this improvement is due to competition that includes the process of cutting the price of maintenance service. Indeed, cutting price leads to attract more demand from airline, resulting in an increase in the net profit. Of course, not all the providers enjoy profit improvement from the competition, as some providers cannot cut their price due to some financial obligations. This interprets why the last provider suffers from the competition game as his profit decreases by around 39%.

Table 5.4: The performance of the first and second models while handling the net profit of maintenance providers

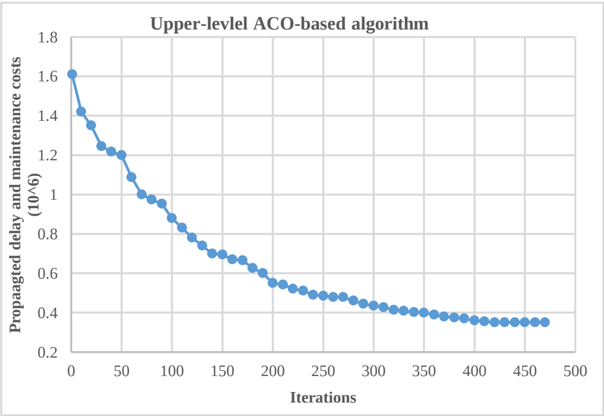
Maintenance provider	First model (Coordination + Competition)	Second model (Only Coordination)	Improvement (%)
Provider 1	106,100	56,982	86.19
Provider 2	64,082	62,162	3.08
Provider 3	57,431	46,622	23.18
Provider 4	34,931	56,982	-38.69

To summarize, considering the competition through the NG is fruitful for the airline as it causes a reduction in the maintenance cost. Meanwhile, this competition is also useful for majority of the providers as it helps to attract more demand from the airline, resulting in an increase in the net profit.

5.6.6 Importance of the LFSG

Similar to the previous section, we need to compare between two situations; considering the coordination between the airline and maintenance providers and neglecting this coordination. The first situation can be captured by the first model proposed in the previous section. The second situation, on the other hand, can be represented by a model that only considers the competition through the NG, while neglecting the coordination, meaning a separate FDARP of airline and MSP of maintenance providers. We call the model for this second situation, the third model, in which the FDARP of airline can be represented using Eqs. (5.1) -(5.7), while neglecting the linkage constraints. The MSP of maintenance providers of the third model can be represented using Eqs. (5.8) -(5.12), while redesigning the first constraints in (5.8) to be $RTAM_{fm} = SDT_{fm}$. Finally, the competition captured by the third model can be represented using Eqs. (5.13) -(5.15). To solve the third model, we can follow this procedure. Firstly, the FDARP and the MSP can be solved by using the upper and lower-level ACO-based algorithms, respectively. Secondly, for the competition part, it can be handled using the analytical method explained in section 5.4.2.

The results of the upper and lower-level ACO-based algorithms are shown in Figure 5.7, in which the upper-level algorithm converges after 470 iterations and returns its best value of 351,000 for the airline, whereas the lower-level algorithm converges after 500 iterations and achieves its best value of 265,705 for the maintenance providers.



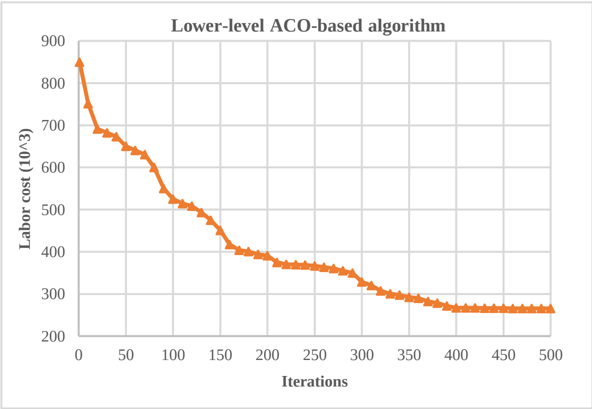


Figure 5.7: Convergence of the upper and lower-level ACO-based algorithms

Table 5.5 summarizes the performance results of the first and third models, while handling the airline and maintenance providers costs. In a close look at the results, we can notice outperformance for the first model the third model by about 9.40%, and 22%, while handling the airline and maintenance providers costs, respectively. The rationale behind this outperformance is because considering the coordination between the airline and maintenance providers gives both players the opportunity to keep adjusting their routing and staffing decisions to improve their results. This finally leads to a reduction in the propagated delay and labor costs paid by airline and maintenance providers, respectively. In contrast to the first model, the third model neglects the coordination, so that the airline and maintenance providers lose the opportunity of adjusting their routing and staffing decisions, leading finally to higher costs for airline and maintenance providers.

Table 5.5: The performance of the first and third models while handling the airline and maintenance providers costs

Costs	First model (Coordination + Competition)	Third model (Only competition)	Outperformance (%)
Propagated delay and maintenance costs by airline	318,000	351,000	9.40
Labor cost by maintenance providers	207,500	265,705	22

In Table 5.6, we report the results of the two models in handling the net profit of the maintenance providers. It is noticed from Table 5.6 that all the providers enjoy better profits using the first model instead of using the third model which neglects the coordination. This is mainly due to the coordination that helps the maintenance providers to minimize the labor costs, which results in increasing the net profit.

Table 5.6: The performance of the first and third models while handling the net profit of maintenance providers

Maintenance provider	First model (Coordination + Competition)	Third model (Only competition)	Improvement (%)
Provider 1	106,100	80,271	32.17
Provider 2	64,082	50,488	26.92
Provider 3	57,431	45,196	27.07
Provider 4	34,931	28,134	24.15

In a conclusion, considering the coordination between the airline and maintenance providers through the LFSG is important for the airline as it causes a reduction in the propagated delay cost. On the other hand, the maintenance providers can benefit from the coordination as it helps to minimize the labor cost, resulting in an increase in the net profit.

5.6.7 Performance analysis

The performance of the SNGM, and the importance of the LFSG and NG have been presented in the previous sections. Presenting the SNGM performance is not enough in demonstrating its importance over the existing models in the literature. In this connection, we extend our experiments to compare the proposed model performance with traditional models in which the coordination and competition are not considered, named as the fourth model. Our proposed model is the same as the first model presented in the previous two sections, whereas the fourth model is similar to the third model presented in section 5.6.6, except neglecting the competition captured by the NG.

Figure 5.8 illustrates the results of the upper and lower-level ACO-based algorithms. By looking to the graphs, we can see that both algorithms converge after 300 iterations and returns their best value of 430,250 for the airline and 265,705 for the maintenance providers.

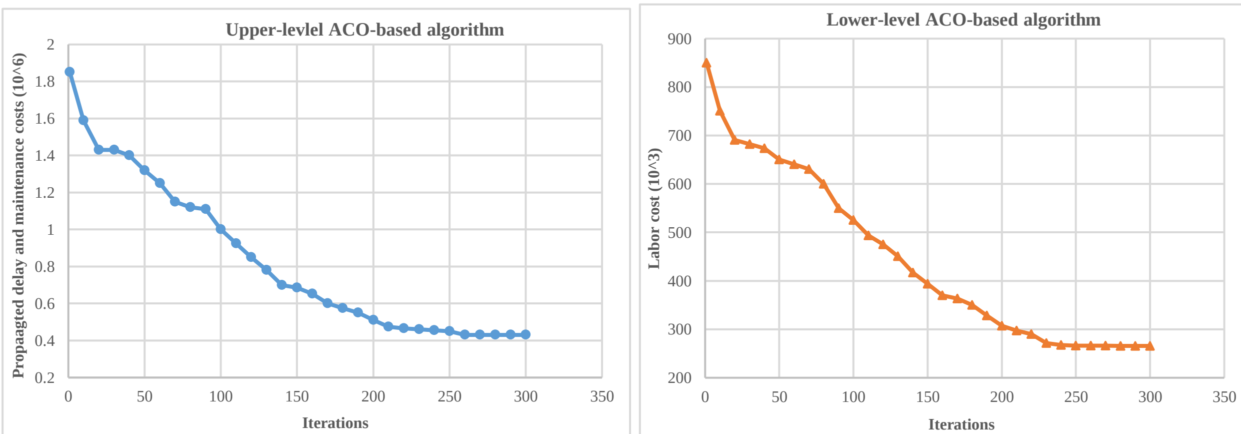


Figure 5.8: Convergence of the upper and lower-level ACO-based algorithms

Table 5.7 compares the performance of the first and fourth models in handling the airline and maintenance providers costs. It is observed in Table 8 that the first model shows better performance over the fourth model by about 26.10 % and 22%, while handling the airline and maintenance providers costs, respectively. The reason for this behavior is due to consideration of the coordination and the competition games. Thanks to the coordination between the airline and maintenance providers as it helps both players to keep adjusting their routing and staffing decisions, so that airline and maintenance providers can get lower propagated delay and labor costs, respectively. The competition game, on the other hand, offers the airline the option to select the cheaper provider, resulting in a reduction in the maintenance cost paid airline. In contrast to the first model, the fourth model neglects both the coordination and the competition, so that the airline and maintenance providers lose all the opportunities explained before, resulting in higher incurred costs for airline and maintenance providers.

Table 5.7: The performance of the first and fourth models while handling the airline and maintenance providers costs

Costs	First model (Coordination + Competition)	Fourth model (No Coordination + No Competition)	Outperformance (%)
Propagated delay and maintenance costs by airline	318,000	430,250	26.10
Labor cost by maintenance providers	207,500	265,705	22

Table 5.8 reports the results of the two models, while handling the net profit of the maintenance providers. It is clear from Table 5.8 that three out of four providers enjoy better profits while adopting the first model in place of adopting the fourth model. The better profits are gained because of considering the coordination that causes a reduction in the labor cost and respecting the competition that leads to attract more demand. These two points leads finally to improve the net profit of the providers. As explained earlier, some providers cannot reduce their prices due to some financial restrictions, so that they cannot attract more demand, leading to a reduction in the net profit, as shown in the case of provider 4.

Table 5.8: The performance of the first and fourth models while handling the net profit of maintenance providers

Maintenance provider	First model (Coordination + Competition)	Fourth model (No Coordination + No Competition)	Improvement (%)
Provider 1	106,100	42,054	152.29
Provider 2	64,082	45,849	39.76
Provider 3	57,431	34,387	67.01
Provider 4	34,931	42,028	-16.88

In conclusion, considering the coordination between the airline and maintenance providers through the LFSG and the competition among maintenance providers through the NG, are important for airline and maintenance providers. For airline, it enjoys lower propagated delay cost owing to the coordination and benefits with a lower maintenance cost due to the competition. For maintenance providers, they gain lower labor cost due to the coordination, whereas the net profit of majority of providers are improved while considering the competition.

In this study, we propose a neural network-based algorithm in order to accurately forecast the NPD. To demonstrate the importance of this algorithm, we extend our experiments to compare the performance of the proposed algorithm with the expected value approach. The experiments include forecasting the NPD for the collected data using the two methods, then using the forecasted NPD to solve the proposed SNGM. The results are summarized in Table 5.9. A close look at the results in Table 5.9 shows that the performance of the neural network-based algorithm is more accurate compared to other method. This is because of considering more factors that affect the NPD, including the bad weather, the season, and the maintenance station congestion factors. In addition, the neural network-based algorithm outperforms the other method by about 7.82%, while handling the airline cost. The reason behind this outperformance is that the expected value approach underestimates the NPD, so that the delay is easily propagated, resulting in a higher propagated delay cost, which in turn leads to an increase in the airline cost. The neural network-based algorithm, on the other hand, provides an accurate NPD, so that the delay propagation is avoided, and its related cost is minimized, causing finally in a reduction in the airline cost.

Table 5.9: Results obtained by different forecasting method

Output	Neural network-based algorithm	Expected value approach	Improvement (%)
Root mean square error (RMSE)	9.208	27.26	66.22
Airline cost	318,000	345,000	7.82

5.6.8 Managerial implications

From all the previous discussions, we can figure out some managerial implications as follows:

- The airline can benefit from the LFSG by obtaining lower propagated delay cost due to holding the dominating position in the game, as being the leader in the game. The maintenance providers, on the other hand, enjoy less advantage from the LFSG by getting lower labor cost, owing to their dominated position as acting the followers in the game.
- The competition among the maintenance providers captured by NG should be in favor of airline as it provides the opportunity to select cheaper providers, resulting in a lower

maintenance cost paid by airline. However, this game is also favorable to majority of providers, as it helps them to attract more demand from airline, leading to better net profit. Meanwhile, this game is not preferred by those providers who cannot reduce their prices, as they suffer from lower demand and lower net profit.

- The LFSG is more welcomed to be applied by the maintenance providers than the NG. This stems from the fact that all the providers enjoy lower labor cost owing to the LFSG, as shown in Table 5.6, whereas some providers suffer from lower profit due to the NG, as shown in Table 5.4.
- The change in the maintenance service prices by maintenance providers has an impact on the airline's decisions, as airline changes its routing plan and enjoys with about 8% reduction in the maintenance cost. Meanwhile, this change has a significant impact on the net profit for the maintenance providers, as shown by the significant increase in the net profit of the first and third providers in our case study. These results are shown in Table 5.4.
- Our iterative game algorithm can find the overall Nash equilibrium for the model within 20 minutes. This computational time is acceptable in practice; therefore, our algorithm can be implemented in real industry.
- Data analytics is an important tool for airlines, as it helps in considering massive information, which in turn results in an accurate non-propagated delay forecasting.

5.7 Summary

In this chapter, we discuss how airline and maintenance providers interact with each other in order to maximize their own profit. This situation is captured by proposing an SNGM, consisting of LFSG between the airline and maintenance providers and NG between maintenance providers. We develop an iterative game algorithm in order to find the overall Nash equilibrium for the proposed model.

Towards the goal of verifying the superiority of the proposed model, we use the case study presented in the previous chapter. To conduct this case study, it is necessary to forecast the non-propagated delay (NPD) for airline and the demand-price function for each maintenance providers. For this purpose, we exploit data analytics by developing a neural network-based algorithm to forecast an accurate NPD of one-year data, such that both the historical flight delay data and other external factors like bad weather and maintenance station congestion, are considered. On the other hand, we adopt a data analytics tool, called multiple linear regression algorithm, to predict the relationship between the demand and price for each maintenance provider. The results of the case study reveal significant saving for airline and maintenance providers owing to the LFSG, whereas the NG improves the net profits for majority of the maintenance providers.

Chapter 6 - Conclusions and Future Work

6.1 Conclusions

Aviation and airline maintenance providers are among the most significant worldwide industries. In 2014, it was reported by International Air Transport Association (IATA) that the world fleet is about 24597 aircraft, and the airlines spent around US$62.1 billion on maintenance, which is expected to raise up to US$90 billion by 2024, due to the significant growth in the number of aircraft. Managing that aircraft growth is a difficult task for airlines and maintenance providers. This difficulty stems from the maintenance requirements which are performed by many maintenance providers. Moreover, the aircraft growth will increase the number of maintenance visits by aircraft, leading to congestion at the maintenance providers' hangers. In July 2017, it was reported by flightstats.com that 25% of the flight delays are due to maintenance congestion. In this regard, AMRP is very significant for airlines as it builds the routes for their aircraft and schedules the maintenance visits. Meanwhile, for maintenance providers, maintenance staffing problem (MSP) is recognized as an effective mean to manage their workforce capacity required to serve the airlines' aircraft.

Although AMRP and MSP have been extensively studied in the literature, there are still some research gaps that exist in the existing literature. These gaps can be summarized as follows:

1 Most of the OARP models were formulated based on the set-partitioning formulation. This means that the number of feasible routes grows exponentially with the number of

flight legs. Indeed, the weakness of this formulation is that it needs generating an exponential number of feasible routes, which result in disability of this formulation to handle large scale problems.

2 The majority of OARP studies considered some operational maintenance restrictions and neglected the rest. For instance, the restrictions regarding the maximum flying hours restrictions, the maximum number of take-offs, and one maintenance visits for every four days have been considered only on the models by Barnhart et al. (1998) and Haouari et al. (2012). However, the drawbacks of these studies is overlooking the working times and workforce capacity of maintenance stations, except the work by Haouari et al. (2012) that considered the workforce capacity of maintenance stations, but this consideration is relaxed in their computational experiments.

3 Most of FDARP studies are anchored on the expected value of the non-propagated delay. It should be noted here that the non-propagated delay happens due to some external factors, including airport congestion, passenger delays, and bad weather. The pitfall of the expected value approach is that the realized value of the delay for some flights turn out to be significantly different from their expected value, due to the high uncertainty of the delay.

4 One of the glaring facts in the literature is solving FDARP of airlines and MSP of maintenance providers independently, even though there is a clear interdependence between them. Therefore, the routing and staffing plans cannot be operated as planned.

5 Lastly, the price competition among the maintenance providers and its effect on the routing plan of airlines has not been investigated in the literature.

These research gaps motivate us to conduct this research work in three main stages. The first stage of this research work is presented in chapter 3, in which the first two research gaps are covered. Actually, in chapter 3, a new MILP model for OARP is presented, in which all the operational maintenance restriction mandated by FAA, the maintenance stations workforce capacity and the maintenance stations working hours, are taken into consideration. In addition, to solve the presented model, an effective solution algorithm is proposed. Moreover, we modify the proposed model with the aim of assessing implications on profitability after taken into consideration the maintenance stations workforce capacity.

To solve our proposed model, first, a commercial software called CPLEX is adopted. Actually, CPLEX provides optimal solutions for small test cases, but feasible solutions cannot be provided for medium and large test instances. Towards the goal of handling medium and large test instances, the proposed solution algorithm is adopted which shows a good performance while solving different sizes of test instances. For small-scale test instances, the best solutions reach the exact solutions, whereas the average solutions deviate by at most 0.83% from exact solutions. With respect to the computational time, a fast performance for the proposed solution algorithm is shown as it can produce the solution in about 3 seconds, whereas 2.5 hours is taken by CPLEX to solve the same problem. For large-scale test instances, the best solutions provided by the proposed algorithm reach the upper bound, whereas the average solutions deviate by at most 0.66%. from the upper bound. The fast performance of the proposed algorithm is also noticed while solving the large-scale test, as it can solve these cases in a few minutes. It is interesting to mention that the proposed algorithm is tested to solve a test instance that is larger than the size of the largest fleet in the world, named Southwest Airline Boeing 737-700, which include 350 aircraft to cover 3469 flight legs in 4 days. The results show that the best solution reaches the upper bound, whereas the average solution deviates by 0.83%. from the upper bound. These results are achieved within 35 minutes, which is a short computational time.

The experiments in chapter 3 are extended for two objectives. First objective is to benchmark the performance of the proposed solution algorithm with the existing solution methods, like CA. By doing so, the results reveal an outperformance of the proposed algorithm over CA, in different two aspects; the solution quality and the computational time. Second objective is to assess the implication on the profitability after considering the workforce capacity of maintenance stations. The results demonstrate an increase in the profitability by about 18.78% for the largest case after considering the maintenance stations workforce capacity.

Later, we start the second stage of our research work, as shown in chapter 4 with the objective to fulfill the third and fourth research gaps. In chapter 4, we propose a scenario-based stochastic FDARP in order to find out better representation for the non-propagated delay. In addition, a bi-level optimization model for coordinated decision support system of scenario-based stochastic FDARP and MSP is proposed by utilizing the Stackelberg game. In this game, scenario-based stochastic FDARP, which is solved by the airline, acts as a leader with the objective of minimizing the expected propagated delay cost. On the other hand, MSP, which is handled by the maintenance providers, plays the role of the follower that responds rationally to the decisions taken by the leader regarding the real departure time of the airline's aircraft from the maintenance providers' stations. Towards the goal solving the developed bi-level optimization model and achieving the Stackelberg equilibrium, a nested ACO-based algorithm is proposed.

To validate the superiority of the proposed model, a case study of the proposed model for a Middle Eastern major airline and four maintenance providers is presented. The results of the case study reveal that both of airline and maintenance providers achieve significant savings in their operational costs, if compared to the results obtained from the traditional non-joint optimization method. This indicates that the proposed model has great potential for implementation in actual practice.

Although chapter 4 presents a formulation for a unique problem in the literature, there are two main limitations for the proposed model. Firstly, it overlooks the price competition existed among maintenance providers. which can easily interrupt the routing plan constructed by the FDARP. Secondly, in chapter 4, the scenario-based stochastic framework for FDARP only focuses on analyzing the historical data and overlooks some other features that affect the delays, including the bad weather, peak seasons, and maintenance station congestion. So, for better forecasting the non-propagated delay, the above described features should be taken into consideration besides the historical data.

The aforementioned two limitations are considered in the last stage of this research work, as presented in chapter 5, in which we propose an SNGM. This model captures the interdependence between FDARP of airline and MSP of maintenance providers in the presence of the price competition among maintenance providers. In particular, SNGM consists of two sub-games: LFSG between the airline and maintenance providers and NG between maintenance providers. We develop an iterative game algorithm in order to find the overall Nash equilibrium for the proposed model.

Towards the goal of verifying the superiority of the proposed model, we use the case study presented in the chapter 4. To conduct this case study, it is necessary to forecast the non-propagated delay (NPD) for airline and the demand-price function for each maintenance providers. For this purpose, we exploit data analytics by developing a neural network-based algorithm to forecast an accurate NPD of one-year data, such that both the historical flight delay data and other external factors like bad weather and maintenance station congestion, are considered. On the other hand, we adopt a data analytics tool, called multiple linear regression algorithm, to predict the relationship between the demand and price for each maintenance provider. The results of the case study reveal significant saving for airline and maintenance providers owing to the LFSG, whereas the NG improves the net profits for majority of the maintenance providers.

6.2 Future Work

Although this research work proposes new models for airlines and maintenance providers, there are some limitations that are suggested for future work. These suggestions can be summarized as follows:

- The scope of the routing model is limited to 4-day planning horizon. It would be beneficial to extend 4-day planning horizon to be weekly planning horizon, in which the number of flight legs and aircraft increase significantly. In addition, designing a solution method to handle this model would be another research direction.

- The MSP proposed in this research work assumes deterministic workforce capacity. In this regard, relaxing this assumption to be stochastic is suggested for future work.
- In the research work, we discussed the interdependence between FDARP and MSP. In real practice, the maintenance providers receive the demand from the airlines, and then use it as an input to determine the staffing plan by solving the MSP problem. Based on the staffing plan, the maintenance providers build the working load or roster for each individual worker by solving the maintenance rostering problem (MRP). From this description, we can imagine that any disruption to the routing plan due to flight delays will cause interruption to the staffing plan, and subsequently the rostering plan will be interrupted. This would result in cancelling the working loads of some workers or adding extra work to others, leading to unstable working loads of the workers of the maintenance providers. Therefore, the integration of MSP and MRP for the maintenance providers is imperative to avoid any disruption that comes from the routing plan of the airlines. This integration would be a promising idea for future research.
- In this research, we discussed the interdependence between airlines and maintenance providers, while considering the competition among the maintenance providers. On a trial to approach the reality, it would be fruitful to extend this research to consider the competition between the airlines.

References

ABARA, J. 1989. Applying Integer Linear Programming to the Fleet Assignment Problem. *Interfaces,* 19**,** 20-28.

ACHOUR, H., GAMACHE, M., SOUMIS, F. & DESAULNIERS, G. 2007. An Exact Solution Approach for the Preferential Bidding System Problem in the Airline Industry. *Transportation Science,* 41**,** 354-365.

ANANDALINGAM, G. & WHITE, D. J. 1990. A solution method for the linear static Stackelberg problem using penalty functions. *IEEE Transactions on Automatic Control,* 35**,** 1170-1173.

ANBIL, R., GELMAN, E., PATTY, B. & TANGA, R. 1991. Recent Advances in Crew-Pairing Optimization at American Airlines. *Interfaces,* 21**,** 62-74.

BALSEIRO, S. R., LOISEAU, I. & RAMONET, J. 2011. An Ant Colony algorithm hybridized with insertion heuristics for the Time Dependent Vehicle Routing Problem with Time Windows. *Computers & Operations Research,* 38**,** 954-966.

BARD, J. F. & FALK, J. E. 1982. An explicit solution to the multi-level programming problem. *Computers & Operations Research,* 9**,** 77-100.

BARNHART, C., BOLAND, N. L., CLARKE, L. W., JOHNSON, E. L., NEMHAUSER, G. L. & SHENOI, R. G. 1998. Flight String Models for Aircraft Fleeting and Routing. *Transportation Science,* 32**,** 208-220.

BARNHART, C., HATAY, L. & JOHNSON, E. L. 1995. Deadhead Selection for the Long-Haul Crew Pairing Problem. *Operations Research,* 43**,** 491-499.

BARNHART, C., KNIKER, T. S. & LOHATEPANONT, M. 2002. Itinerary-Based Airline Fleet Assignment. *Transportation Science,* 36**,** 199-217.

BAŞDERE, M. & BILGE, Ü. 2014. Operational aircraft maintenance routing problem with remaining time consideration. *European Journal of Operational Research,* 235**,** 315-328.

BATLEY, R., DARGAY, J. & WARDMAN, M. 2011. The impact of lateness and reliability on passenger rail demand. *Transportation Research Part E: Logistics and Transportation Review,* 47**,** 61-72.

BEAUMONT, N. 1997. Scheduling staff using mixed integer programming. *European Journal of Operational Research,* 98**,** 473-484.

BÉLANGER, N., DESAULNIERS, G., SOUMIS, F., DESROSIERS, J. & LAVIGNE, J. 2006. Weekly airline fleet assignment with homogeneity. *Transportation Research Part B: Methodological,* 40**,** 306-318.

BERGE, M. E. & HOPPERSTAD, C. A. 1993. Demand Driven Dispatch: A Method for Dynamic Aircraft Capacity Assignment, Models and Algorithms. *Operations Research,* 41**,** 153-168.

BURKE, E. K., DE CAUSMAECKER, P., DE MAERE, G., MULDER, J., PAELINCK, M. & VANDEN BERGHE, G. 2010. A multi-objective approach for robust airline scheduling. *Computers & Operations Research,* 37**,** 822-832.

CACCHIANI, V. & SALAZAR-GONZÁLEZ, J.-J. 2013. A heuristic approach for an integrated fleet-assignment, aircraft-routing and crew-pairing problem. *Electronic Notes in Discrete Mathematics,* 41**,** 391-398.

CACCHIANI, V. & SALAZAR-GONZÁLEZ, J.-J. 2016. Optimal Solutions to a Real-World Integrated Airline Scheduling Problem. *Transportation Science*.

CAI, G., ZHANG, Z. G. & ZHANG, M. 2009a. Game theoretical perspectives on dual-channel supply chain competition with price discounts and pricing schemes. *International Journal of Production Economics,* 117**,** 80-96.

CAI, X., SUN, X. & ZHOU, X. 2004. Stochastic scheduling subject to machine breakdowns: The preemptive-repeat model with discounted reward and other criteria. *Naval Research Logistics (NRL),* 51**,** 800-817.

CAI, X., WU, X. & ZHOU, X. 2009b. Stochastic Scheduling Subject to Preemptive-Repeat Breakdowns with Incomplete Information. *Operations Research,* 57**,** 1236-1249.

CAI, X. & ZHOU, S. 1999. Stochastic Scheduling on Parallel Machines Subject to Random Breakdowns to Minimize Expected Costs for Earliness and Tardy Jobs. *Operations Research,* 47**,** 422-437.

CAPPANERA, P. & GALLO, G. 2004. A Multicommodity Flow Approach to the Crew Rostering Problem. *Operations Research,* 52**,** 583-596.

CHEN, Z.-L., LI, S. & TIRUPATI, D. 2002. A scenario-based stochastic programming approach for technology and capacity planning. *Computers & Operations Research,* 29**,** 781-806.

CHOI, S. C. 1991. Price Competition in a Channel Structure with a Common Retailer. *Marketing Science,* 10**,** 271-296.

CHU, H. D., GELMAN, E. & JOHNSON, E. L. 1997. Solving large scale crew scheduling problems. *European Journal of Operational Research,* 97**,** 260-268.

CLARKE, L., JOHNSON, E., NEMHAUSER, G. & ZHU, Z. 1997. The aircraft rotation problem. *Annals of Operations Research,* 69**,** 33-46.

COHN, A. M. & BARNHART, C. 2003. Improving Crew Scheduling by Incorporating Key Maintenance Routing Decisions. *Operations Research,* 51**,** 387-396.

CORDEAU, J.-F., STOJKOVIĆ, G., SOUMIS, F. & DESROSIERS, J. 2001. Benders Decomposition for Simultaneous Aircraft Routing and Crew Scheduling. *Transportation Science,* 35**,** 375-388.

DAWID, H., KÖNIG, J. & STRAUSS, C. 2001. An enhanced rostering model for airline crews. *Computers & Operations Research,* 28**,** 671-688.

DAY, P. R. & RYAN, D. M. 1997. Flight Attendant Rostering for Short-Haul Airline Operations. *Operations Research,* 45**,** 649-661.

DESAULNIERS, G., DESROSIERS, J., DUMAS, Y., MARC, S., RIOUX, B., SOLOMON, M. M. & SOUMIS, F. 1997a. Crew pairing at Air France. *European Journal of Operational Research,* 97**,** 245-259.

DESAULNIERS, G., DESROSIERS, J., DUMAS, Y., SOLOMON, M. M. & SOUMIS, F. 1997b. Daily Aircraft Routing and Scheduling. *Management Science,* 43**,** 841-855.

DÍAZ-RAMÍREZ, J., HUERTAS, J. I. & TRIGOS, F. 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.

DIETZ , D. C. & ROSENSHINE , M. 1997. Optimal specialization of a maintenance workforce. *IIE Transactions,* 29**,** 423-433.

DONG, Z., CHUHANG, Y. & LAU, H. Y. K. H. 2016. An integrated flight scheduling and fleet assignment method based on a discrete choice model. *Computers & Industrial Engineering,* 98**,** 195-210.

DUMAS, J., AITHNARD, F. & SOUMIS, F. 2009. Improving the objective function of the fleet assignment problem. *Transportation Research Part B: Methodological,* 43**,** 466-475.

DUNBAR, M., FROYLAND, G. & WU, C.-L. 2014. An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. *Computers & Operations Research,* 45**,** 68-86.

EHRGOTT, M. & RYAN, D. M. 2002. Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis,* 11**,** 139-150.

ESMAEILI, M., ARYANEZHAD, M.-B. & ZEEPHONGSEKUL, P. 2009. A game theory approach in seller–buyer supply chain. *European Journal of Operational Research,* 195**,** 442-448.

FAHLE, T., JUNKER, U., KARISCH, S., KOHL, N., SELLMANN, M. & VAABEN, B. 2002. Constraint Programming Based Column Generation for Crew Assignment. *Journal of Heuristics,* 8**,** 59-81.

FARKAS, A. 1995. *The influence of network effects and yield management on airline fleet assignment decisions.* Ph.D. Dissertation, Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA.

FEO, T. A. & BARD, J. F. 1989. Flight Scheduling and Maintenance Base Planning. *Management Science,* 35**,** 1415-1432.

FORTUNY-AMAT, J., XE & MCCARL, B. 1981. A Representation and Economic Interpretation of a Two-Level Programming Problem. *The Journal of the Operational Research Society,* 32**,** 783-792.

GAMACHE, M., SOUMIS, F., MARQUIS, G. & DESROSIERS, J. 1999. A Column Generation Approach for Large-Scale Aircrew Rostering Problems. *Operations Research,* 47**,** 247-263.

GAMACHE, M., SOUMIS, F., VILLENEUVE, D., DESROSIERS, J. & GÉLINAS, É. 1998. The Preferential Bidding System at Air Canada. *Transportation Science,* 32**,** 246-255.

GHOFRANI, F., HE, Q., GOVERDE, R. M. P. & LIU, X. 2018. Recent applications of big data analytics in railway transportation systems: A survey. *Transportation Research Part C: Emerging Technologies,* 90**,** 226-246.

GOPALAN, R. & TALLURI, K. T. 1998. The Aircraft Maintenance Routing Problem. *Operations Research,* 46**,** 260-271.

GRAVES, G. W., MCBRIDE, R. D., GERSHKOFF, I., ANDERSON, D. & MAHIDHARA, D. 1993. Flight Crew Scheduling. *Management Science,* 39**,** 736-745.

GUO, Y., MELLOULI, T., SUHL, L. & THIEL, M. P. 2006. A partially integrated airline crew scheduling approach with time-dependent crew capacities and multiple home bases. *European Journal of Operational Research,* 171**,** 1169-1181.

GÜRKAN, H., GÜREL, S. & AKTÜRK, M. S. 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., BARNHART, C., JOHNSON, E., MARSTEN, R., NEMHAUSER, G. & SIGISMONDI, G. 1995. The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming,* 70**,** 211-232.

HAOUARI, M., AISSAOUI, N. & MANSOUR, F. Z. 2009. Network flow-based approaches for integrated aircraft fleeting and routing. *European Journal of Operational Research,* 193**,** 591-599.

HAOUARI, M., SHAO, S. & SHERALI, H. D. 2012. A Lifted Compact Formulation for the Daily Aircraft Maintenance Routing Problem. *Transportation Science,* 47**,** 508-525.

HAOUARI, M., SHERALI, H., MANSOUR, F. & AISSAOUI, N. 2011. Exact approaches for integrated aircraft fleeting and routing at TunisAir. *Computational Optimization & Applications,* 49**,** 213-239.

HASHIM, H. A. & ABIDO, M. A. 2015. Fuzzy Controller Design Using Evolutionary Techniques for Twin Rotor MIMO System: A Comparative Study. *Computational Intelligence and Neuroscience,* 2015**,** 11.

HASHIM, H. A., AYINDE, B. O. & ABIDO, M. A. 2016. Optimal placement of relay nodes in wireless sensor network using artificial bee colony algorithm. *Journal of Network and Computer Applications,* 64**,** 239-248.

HASHIM, H. A., BROWN, L. J. & MCISAAC, K. Nonlinear Explicit Stochastic Attitude Filter on SO(3). in Proceedings of the 57th IEEE Conference on Decision and Control (CDC), 2018a. pp. 1-7.

HASHIM, H. A., BROWN, L. J. & MCISAAC, K. 2018b. Nonlinear Stochastic Attitude Filters on the Special Orthogonal Group 3: Ito and Stratonovich. *IEEE Transactions on Systems, Man, and Cybernetics: Systems***,** 1-13.

HASHIM, H. A., BROWN, L. J. & MCISAAC, K. 2018c. Nonlinear Stochastic Position and Attitude Filter on the Special Euclidean Group 3. *Journal of the Franklin Institute***,** pp. 1–27 (In press).

HASHIM, H. A., EL-FERIK, S. & ABIDO, M. A. 2015. A fuzzy logic feedback filter design tuned with PSO for adaptive controller. *Expert Systems with Applications,* 42**,** 9077-9085.

HOFFMAN, K. L. & PADBERG, M. 1993. Solving Airline Crew Scheduling Problems by Branch-and-Cut. *Management Science,* 39**,** 657-682.

HSU, C.-W., LEE, Y. & LIAO, C.-H. 2010. Competition between high-speed and conventional rail systems: A game theoretical approach. *Expert Systems with Applications,* 37**,** 3162-3170.

HUANG, S.-H., HUANG, Y.-H., BLAZQUEZ, C. A. & PAREDES-BELMAR, G. 2018. Application of the Ant Colony Optimization in the Resolution of the Bridge Inspection Routing Problem. *Applied Soft Computing*.

JACOBS, T. L., SMITH, B. C. & JOHNSON, E. L. 2008. Incorporating Network Flow Effects into the Airline Fleet Assignment Process. *Transportation Science,* 42**,** 514-529.

JARRAH, A. I., GOODSTEIN, J. & NARASIMHAN, R. 2000. An Efficient Airline Re-Fleeting Model for the Incremental Modification of Planned Fleet Assignments. *Transportation Science,* 34**,** 349-363.

JIANG, H. & BARNHART, C. 2009. Dynamic Airline Scheduling. *Transportation Science,* 43**,** 336-354.

JIANG, H. & BARNHART, C. 2013. Robust airline schedule design in a dynamic scheduling environment. *Computers & Operations Research,* 40**,** 831-840.

KABBANI, N. M. & PATTY, B. W. Aircraft routing at American airlines. In Proceedings of the 32nd annual symposium of AGIFORS, 1992 Budapest, Hungary.

KLABJAN, D., JOHNSON, E. L., NEMHAUSER, G. L., GELMAN, E. & RAMASWAMY, S. 2001. Airline Crew Scheduling with Regularity. *Transportation Science,* 35**,** 359-374.

KLABJAN, D., JOHNSON, E. L., NEMHAUSER, G. L., GELMAN, E. & RAMASWAMY, S. 2002. Airline Crew Scheduling with Time Windows and Plane-Count Constraints. *Transportation Science,* 36**,** 337-348.

LAN, S., CLARKE, J.-P. & BARNHART, C. 2006. Planning for Robust Airline Operations: Optimizing Aircraft Routings and Flight Departure Times to Minimize Passenger Disruptions. *Transportation Science,* 40**,** 15-28.

LEE, L. H., LEE, C. U. & TAN, Y. P. 2007. A multi-objective genetic algorithm for robust flight scheduling using simulation. *European Journal of Operational Research,* 177**,** 1948-1968.

LEVIN, A. 1971. Scheduling and Fleet Routing Models for Transportation Systems. *Transportation Science,* 5**,** 232-255.

LI, C., QI, X. & SONG, D. 2016. Real-time schedule recovery in liner shipping service with regular uncertainties and disruption events. *Transportation Research Part B: Methodological,* 93**,** 762-788.

LIANG, Z. & CHAOVALITWONGSE, W. A. 2012. A Network-Based Model for the Integrated Weekly Aircraft Maintenance Routing and Fleet Assignment Problem. *Transportation Science,* 47**,** 493-507.

LIANG, Z., CHAOVALITWONGSE, W. A., HUANG, H. C. & JOHNSON, E. L. 2011. On a New Rotation Tour Network Model for Aircraft Maintenance Routing Problem. *Transportation Science,* 45**,** 109-120.

LIANG, Z., FENG, Y., ZHANG, X., WU, T. & CHAOVALITWONGSE, W. A. 2015. Robust weekly aircraft maintenance routing problem and the extension to the tail assignment problem. *Transportation Research Part B: Methodological,* 78**,** 238-259.

LIU, B. 1998. Stackelberg-Nash equilibrium for multilevel programming with multiple followers using genetic algorithms. *Computers & Mathematics with Applications,* 36**,** 79-89.

LUČIC, P. & TEODOROVIC, D. 1999. Simulated annealing for the multi-objective aircrew rostering problem. *Transportation Research Part A: Policy and Practice,* 33**,** 19-45.

MAENHOUT, B. & VANHOUCKE, M. 2010. A hybrid scatter search heuristic for personalized crew rostering in the airline industry. *European Journal of Operational Research,* 206**,** 155-167.

MAHATO, D. P., SINGH, R. S., TRIPATHI, A. K. & MAURYA, A. K. 2017. On scheduling transactions in a grid processing system considering load through Ant Colony Optimization. *Applied Soft Computing,* 61**,** 875-891.

MERCIER, A., CORDEAU, J.-F. & SOUMIS, F. 2005. A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research,* 32**,** 1451-1476.

MERCIER, A. & SOUMIS, F. 2007. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research,* 34**,** 2251-2265.

MISIUNAS, N., OZTEKIN, A., CHEN, Y. & CHANDRA, K. 2016. DEANN: A healthcare analytic methodology of data envelopment analysis and artificial neural networks for the prediction of organ recipient functional status. *Omega,* 58**,** 46-54.

MOHAMMADI, S., SOLEYMANI, S. & MOZAFARI, B. 2014. Scenario-based stochastic operation management of MicroGrid including Wind, Photovoltaic, Micro-Turbine, Fuel Cell and Energy Storage Devices. *International Journal of Electrical Power & Energy Systems,* 54**,** 525-535.

MUÑOZ, M. M. & ABDELAZIZ, F. B. 2012. Satisfactory solution concepts and their relations for Stochastic Multiobjective Programming problems. *European Journal of Operational Research,* 220**,** 430-442.

MURALI, P., DESSOUKY, M., ORDÓÑEZ, F. & PALMER, K. 2010. A delay estimation technique for single and double-track railroads. *Transportation Research Part E: Logistics and Transportation Review,* 46**,** 483-495.

MUTER, İ., İLKER BIRBIL, Ş., BÜLBÜL, K., ŞAHIN, G., YENIGÜN, H., TAŞ, D. & TÜZÜN, D. 2013. Solving a robust airline crew pairing problem with column generation. *Computers & Operations Research,* 40**,** 815-830.

OZDEMIR, H. T. & MOHAN, C. K. 2001. Flight graph based genetic algorithm for crew scheduling in airlines. *Information Sciences,* 133**,** 165-173.

PILLA, V. L., ROSENBERGER, J. M., CHEN, V., ENGSUWAN, N. & SIDDAPPA, S. 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**,** 162-171.

REXING, B., BARNHART, C., KNIKER, T., JARRAH, A. & KRISHNAMURTHY, N. 2000. Airline Fleet Assignment with Time Windows. *Transportation Science,* 34**,** 1-20.

ROSENBERGER, J. M., JOHNSON, E. L. & NEMHAUSER, G. L. 2004. A Robust Fleet-Assignment Model with Hub Isolation and Short Cycles. *Transportation Science,* 38**,** 357-368.

RUSHMEIER, R. A. & KONTOGIORGIS, S. A. 1997. Advances in the Optimization of Airline Fleet Assignment. *Transportation Science,* 31**,** 159-169.

RYAN, D. M. 1992. The Solution of Massive Generalized Set Partitioning Problems in Aircrew Rostering. *The Journal of the Operational Research Society,* 43**,** 459-467.

SADDOUNE, M., DESAULNIERS, G., ELHALLAOUI, I. & SOUMIS, F. 2011. Integrated airline crew scheduling: A bi-dynamic constraint aggregation method using neighborhoods. *European Journal of Operational Research,* 212**,** 445-454.

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.
SAMIMI, A., NIKZAD, M. & SIANO, P. 2017. Scenario-based stochastic framework for coupled active and reactive power market in smart distribution systems with demand response programs. *Renewable Energy,* 109**,** 22-40.
SAMUELSON, P. 1947. *Foundations of Economic Analysis*, Foundations of Economic Analysis, Cam-bridge, MA.
SARAC, A., BATTA, R. & RUMP, C. M. 2006. A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research,* 175**,** 1850-1869.
SCHAEFER, A. J., JOHNSON, E. L., KLEYWEGT, A. J. & NEMHAUSER, G. L. 2005. Airline Crew Scheduling Under Uncertainty. *Transportation Science,* 39**,** 340-348.
SHAFIULLAH, G. M., ALI, A. B. M. S., THOMPSON, A. & WOLFS, P. J. 2010. Predicting vertical acceleration of railway wagons using regression algorithms. *Trans. Intell. Transport. Sys.,* 11**,** 290-299.
SHAO, S., SHERALI, H. D. & HAOUARI, M. 2015. A Novel Model and Decomposition Approach for the Integrated Airline Fleet Assignment, Aircraft Routing, and Crew Pairing Problem. *Transportation Science*.
SHEBALOV, S. & KLABJAN, D. 2006. Robust Airline Crew Pairing: Move-up Crews. *Transportation Science,* 40**,** 300-312.
SHERALI, H. D., BAE, K.-H. & HAOUARI, M. 2013. An Integrated Approach for Airline Flight Selection and Timing, Fleet Assignment, and Aircraft Routing. *Transportation Science,* 47**,** 455-476.
SHERALI, H. D., BISH, E. K. & ZHU, X. 2005. Polyhedral Analysis and Algorithms for a Demand-Driven Refleeting Model for Aircraft Assignment. *Transportation Science,* 39**,** 349-366.
SINHA, S. & SARMAH, S. P. 2010. Coordination and price competition in a duopoly common retailer supply chain. *Computers & Industrial Engineering,* 59**,** 280-295.
SKINDEROWICZ, R. 2017. An improved Ant Colony System for the Sequential Ordering Problem. *Computers & Operations Research,* 86**,** 1-17.
SMITH, B. C. & JOHNSON, E. L. 2006. Robust Airline Fleet Assignment: Imposing Station Purity Using Station Decomposition. *Transportation Science,* 40**,** 497-516.
SOHONI, M., LEE, Y.-C. & KLABJAN, D. 2011. Robust Airline Scheduling Under Block-Time Uncertainty. *Transportation Science,* 45**,** 451-464.
SOUAI, N. & TEGHEM, J. 2009. Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem. *European Journal of Operational Research,* 199**,** 674-683.
SRIRAM, C. & HAGHANI, A. 2003. An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A: Policy and Practice,* 37**,** 29-48.
TABRIZI, S., GHODSYPOUR, S. H. & AHMADI, A. 2017. Modelling three-echelon warm-water fish supply chain: A bi-level optimization approach under Nash–Cournot equilibrium. *Applied Soft Computing*.
TALLURI, K. 1996. Swapping Applications in a Daily Airline Fleet Assignment. *Transportation science,* 30**,** 237-248.
TALLURI, K. T. 1998. The Four-Day Aircraft Maintenance Routing Problem. *Transportation Science,* 32**,** 43-53.
TAVANA, M., ABTAHI, A.-R., DI CAPRIO, D. & POORTARIGH, M. 2018. An Artificial Neural Network and Bayesian Network model for liquidity risk assessment in banking. *Neurocomputing,* 275**,** 2525-2554.
TEKINER, H., BIRBIL, Ş. İ. & BÜLBÜL, K. 2009. Robust crew pairing for managing extra flights. *Computers & Operations Research,* 36**,** 2031-2048.

TIWARI, S., WEE, H. M. & DARYANTO, Y. 2018. Big data analytics in supply chain management between 2010 and 2016: Insights to industries. *Computers & Industrial Engineering,* 115**,** 319-330.
TSAI, T.-H., LEE, C.-K. & WEI, C.-H. 2009. Neural network based temporal feature models for short-term railway passenger demand forecasting. *Expert Systems with Applications,* 36**,** 3728-3736.
VAN HOESEL, S. 2008. An overview of Stackelberg pricing in networks. *European Journal of Operational Research,* 189**,** 1393-1402.
VANCE, P. H., BARNHART, C., JOHNSON, E. L. & NEMHAUSER, G. L. 1997. Airline Crew Scheduling: A New Formulation and Decomposition Algorithm. *Operations Research,* 45**,** 188-200.
WANG, C., MU, D., ZHAO, F. & SUTHERLAND, J. W. 2015. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Computers & Industrial Engineering,* 83**,** 111-122.
WANG, D., DU, G., JIAO, R. J., WU, R., YU, J. & YANG, D. 2016. A Stackelberg game theoretic model for optimizing product family architecting with supply chain consideration. *International Journal of Production Economics,* 172**,** 1-18.
WEIDE, O., RYAN, D. & EHRGOTT, M. 2010. An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research,* 37**,** 833-844.
XIAO, T., CHOI, T.-M. & CHENG, T. C. E. 2014. Product variety and channel structure strategy for a retailer-Stackelberg supply chain. *European Journal of Operational Research,* 233**,** 114-124.
YAN, S. & CHANG, J.-C. 2002. Airline cockpit crew scheduling. *European Journal of Operational Research,* 136**,** 501-511.
YAN, S., TANG, C. H. & FU, T. C. 2008. An airline scheduling model and solution algorithms under stochastic demands. *European Journal of Operational Research,* 190**,** 22-39.
YAN, S., TANG, C. H. & LEE, M. C. 2007. A flight scheduling model for Taiwan airlines under market competitions. *Omega,* 35**,** 61-74.
YAN, S. & TSENG, C. H. 2002. A passenger demand model for airline flight scheduling and fleet routing. *Computers & Operations Research,* 29**,** 1559-1581.
YAN, S., YANG, T.-H. & CHEN, H.-H. 2004. Airline short-term maintenance manpower supply planning. *Transportation Research Part A: Policy and Practice,* 38**,** 615-642.
YAN, S. & YOUNG, H. F. 1996. A decision support framework for multi-fleet routing and multi-stop flight scheduling. *Transportation Research Part A: Policy and Practice,* 30**,** 379-398.
YANG, D., JIAO, J., JI, Y., DU, G., HELO, P. & VALENTE, A. 2015. Joint optimization for coordinated configuration of product families and supply chains by a leader-follower Stackelberg game. *European Journal of Operational Research,* 246**,** 263-280.
YANG, T.-H., YAN, S. & CHEN, H.-H. 2003. An airline maintenance manpower planning model with flexible strategies. *Journal of Air Transport Management,* 9**,** 233-239.
YEN, J. W. & BIRGE, J. R. 2006. A Stochastic Programming Approach to the Airline Crew Scheduling Problem. *Transportation Science,* 40**,** 3-14.
YIN, P.-Y. & WANG, J.-Y. 2006. Ant colony optimization for the nonlinear resource allocation problem. *Applied Mathematics and Computation,* 174**,** 1438-1453.
YU, Y. & HUANG, G. Q. 2010. Nash game model for optimizing market strategies, configuration of platform products in a Vendor Managed Inventory (VMI) supply chain for a product family. *European Journal of Operational Research,* 206**,** 361-373.
YU, Y., HUANG, G. Q. & LIANG, L. 2009. Stackelberg game-theoretic model for optimizing advertising, pricing and inventory policies in vendor managed inventory (VMI) production supply chains. *Computers & Industrial Engineering,* 57**,** 368-382.

ZEGHAL, F. M., HAOUARI, M., SHERALI, H. D. & AISSAOUI, N. 2011. Flexible aircraft fleeting and routing at "Tunis Air". *The Journal of the Operational Research Society*, 62**,** 368-380.