Master's degree thesis

LOG950 Logistics

Robust integrated models for airline planning

Maryia Hryhoryeva

Number of pages including this page: 84

Molde, 02.06.2020

Molde University College
Specialized University in Logistics

Mandatory statement

Each student is responsible for complying with rules and regulations that relate to examinations and to academic work in general. The purpose of the mandatory statement is to make students aware of their responsibility and the consequences of cheating. Failure to complete the statement does not excuse students from their responsibility.

*Please complete the mandatory statement by placing a mark **in each box** for statements 1-6 below.*		
1.	**I/we hereby declare that my/our paper/assignment is my/our own work, and that I/we have not used other sources or received other help than mentioned in the paper/assignment.**	☒
2.	**I/we hereby declare that this paper** 1. Has not been used in any other exam at another department/university/university college 2. Is not referring to the work of others without acknowledgement 3. Is not referring to my/our previous work without acknowledgement 4. Has acknowledged all sources of literature in the text and in the list of references 5. Is not a copy, duplicate or transcript of other work	Mark each box: 1. ☒ 2. ☒ 3. ☒ 4. ☒ 5. ☒
3.	I am/we are aware **that any breach of the above will be considered as cheating, and may result in annulment of the examination and exclusion from all universities and university colleges in Norway for up to one year, according to the Act relating to Norwegian Universities and University Colleges, section 4-7 and 4-8 and Examination regulations** section 14 and 15.	☒
4.	**I am/we are aware that all papers/assignments may be checked for plagiarism by a software assisted plagiarism check**	☒
5.	**I am/we are aware that Molde University College will handle all cases of suspected cheating according to prevailing guidelines.**	☒
6.	**I/we are aware of the University College's rules and regulation for using sources**	☒

Personal protection

Personal Data Act

Research projects that processes personal data according to Personal Data Act, should be notified to Data Protection Services (NSD) for consideration.

Have the research project been considered by NSD? ☐yes ☒no

- If yes:

Reference number:

- If no:

I/we hereby declare that the thesis does not contain personal data according to Personal Data Act.: ☒

Act on Medical and Health Research

If the research project is effected by the regulations decided in Act on Medical and Health Research (the Health Research Act), it must be approved in advance by the Regional Committee for Medical and Health Research Ethic (REK) in your region.

Has the research project been considered by REK? ☐yes ☒no

- If yes:

Reference number:

Publication agreement

ECTS credits: 30

Supervisor: Lars Magnus Hvattum, Mohamed Ben Ahmed

Agreement on electronic publication of master thesis

Author(s) have copyright to the thesis, including the exclusive right to publish the document (The Copyright Act §2).

All theses fulfilling the requirements will be registered and published in Brage HiM, with the approval of the author(s).

Theses with a confidentiality agreement will not be published.

I/we hereby give Molde University College the right to, free of charge, make the thesis available for electronic publication: ☒yes ☐no

Is there an agreement of confidentiality? ☐yes ☒no

(A supplementary confidentiality agreement must be filled in)

- If yes:

Can the thesis be online published when the period of confidentiality is expired? ☐yes ☐no

Date: 01.06.2020

Preface

This thesis is a final stage of the two-year Master's program in Logistics Analytics at Molde University College, Norway. The work on the subject of the thesis was continuing from November 2019 till June 2020 and gave me inspiration for future research.

I would like to thank Molde University College for creating proper conditions for this study to be done and all the teachers who gave me the needed knowledge.

Besides, I would like to express my sincere gratitude to my supervisors Lars Magnus Hvattum and Mohamed Ben Ahmed, who were helping me during the whole period of the research.

I am also thankful to my family and friends for their support and encouragement.

Summary

This study examines the possibilities of integration of three airline planning problems: fleet assignment, aircraft routing, and crew pairing while satisfying aircraft maintenance requirements and several crew working rules. Besides, the following robustness techniques are embedded in the model: avoidance of short aircraft connections and stimulation of crews to follow the aircraft on any connection. In this thesis we present the integrated robust mathematical model of those problems, apply the reformulation-linearization technique to obtain a linear equivalent model, perform a programming implementation using AMPL modeling language and show the results of testing the model using a commercial solver on several data instances provided by United Airlines Company.

Content

1. Introduction

1.1 Background of the industry and research problem

Aviation is one of the biggest global industries and contributors to world business in terms of financial turnover. Since the first commercial flight was launched at the beginning of the twentieth century, the aviation sphere of the industry was evolving at a fast pace. In 2019, approximately 4.5 billion passengers were transported by air vehicles. Presently, around 65.5 million people are globally occupied in the airline industry and related spheres. Among those people, 10.2 million have direct jobs in aviation which mean full-time positions created by the airline industry itself (Air Transport Action Group 2018a).

Following the foundation of the airline companies, their management is focused on the optimization of processes in order to increase the revenue and decrease the losses. As the main source of profit for airline companies is a provision of transportation services to people, it is crucial to develop an optimal flying schedule that satisfies the market demand.

Airline operations contain many problems and tasks to be solved. Development of optimal flight schedule requires finding a solution for several sub-problems: schedule design, fleet assignment, aircraft routing, crew pairing, and crew rostering.

Figure 1 - Airline planning problems

As the integration of all problems' models is too complex to be solved with already existing computational means, most often those problems are solved sequentially (Figure 1), when in the model of one problem the optimal solution result from the prior problem is used. This leads to suboptimal solutions of the subsequent problems or even infeasibility of the final generated schedule while the full problem itself is feasible (Papadakos 2007). If the problems are not too voluminous, it could be possible to integrate the models of some of them in one, and thus to avoid suboptimal solutions.

Another aspect that influences on the airline company's profitability is losses. According to Ball et al. (2010) in 2007 air transportation delays in the United States caused 31.2 billion dollars of financial losses, including direct costs to airlines and passengers as well as the impact on the country's GDP. More precise information can be observed in Table 1.

Table 1: Direct cost of US air transportation delay in 2007(Ball et al. 2010)

Cost Component	**Cost ($ billions)**
Costs to Airlines	8.3
Costs to Passengers	16.7
Costs from Lost Demand	2.2
Total Direct Cost	27.2
Impact on GDP	4.0
Total Cost	31.2

Hence, it is crucial for airline companies to avoid the situation of delayed and canceled flights. To reduce the probability of delay and cancellation to occur, it is necessary to introduce several preventive measurements during the stage of flight schedule creation. Those measurements are called robustness criteria and they help to build a well-balanced fleet and crew pairing schedules, resistant to flight delays and cancellations.

Thus, this study is aimed to develop an integrated model of several airline planning problems as well as adding robustness criteria to increase the model's disruption resistance. Hereinafter, the literature overview of the already existing studies in this sphere of research is presented to examine the history of the research problem and to highlight the distinguishing features of our work as well as its newness and relevance.

1.2 Significance of the study

The goal of our study is to create a compact robust integrated model of fleet assignment, aircraft routing, and crew pairing problems, which can produce a fleet and crew pairing schedules avoiding suboptimal decisions and increasing disruption resistance. This helps to prevent delays and cancellations of flights and their propagation through the flight network. As a result, airline companies can gain satisfaction and loyalty of their passengers thanks to the improvement of the provided services and evade situations of financial losses and unreasonable operational costs.

Another issue that is influenced by the optimization of airline operations is carbon dioxide emissions. By optimizing flight routes it is possible to reduce the environmental impact.

Optimization of the airline planning process also reduces the probability of a situation called deadheading, when a company should transport staff as regular passengers for them to start the next duty at the necessary base, which can lead to reputation impairment of the company, as it happened with United Airlines on the 9th of April, 2017 (Victor and Stevens 2017).

The models and ideas utilized in this work could be used not only in the airline industry but also in other spheres of logistics and supply chain management, such as for vehicle routing problems applied to land and air cargo transportation as well as delivery of goods.

1.3 Structure of thesis

Chapter 2. Problem description

Overview of main problems in airline planning and techniques that aimed to reduce the propagation of delays and cancellations within the flight network.

Chapter 3. Research questions & methodology

This chapter introduces the research questions and used methods.

Chapter 4. Literature review

In this chapter, the overview of previously made research is presented.

Chapter 5. Mathematical formulation

Presents the mathematical notation, nonlinear mixed-integer programming model, and its linearization.

Chapter 6. Computational study

This chapter provides the data description and the results of computational experiments.

Chapter 7. Conclusion and future work

Concludes the results of the work and suggests future enhancements.

2. Problem description

One of the main issues that airline companies face is the creation of optimal operational schedules. This includes finding a solution for the next problems: schedule design, fleet assignment, aircraft routing, crew pairing, and crew rostering. As it was mentioned before, solving those problems separately and using the output of one problem as an input for another leads to the suboptimal or infeasible solutions. Thus, in this study fleet assignment, aircraft routing, and crew pairing problems are integrated into one model to archive the optimal solution using disposable computational means. Below, the detailed description of each of the main airline planning problems is presented.

2.1 Main problems in airline planning

The first problem to be solved during the airline planning stage is *schedule design*. It implicates the determination of destinations of flying routes, time to perform the flight, and frequency of flights. The goal of schedule design is to produce a timetable that maximizes the potential revenue according to the customer's demand forecast. Usually, based on the generated schedule all the other airline operations are built. Generally, the timetable for flights is cyclic and repeats itself every day for domestic flights, and every week for international flights (Bazargan 2010). In our case, the data provided by United Airlines Company contains information about the origins and destinations of flights as well as the departure and arrival time. For simplicity, only the domestic flights that repeat themselves every day are used in this study, and the stage of schedule design is not considered further.

The subsequent problem to be solved after schedule design is *fleet assignment*. The goal of the fleet assignment is to match an aircraft type from the possessed fleet with a flight in the schedule (Bazargan 2010). According to Ben Ahmed, Zeghal Mansour, and Haouari (2018), the *aircraft type* as a specific model of aircraft. Aircraft that belong to one type share the same cockpit configuration and the number of seats. On the other hand, *aircraft family* embraces several types of aircraft that have the same cockpit configuration and cockpit rating. For example, aircraft types Airbus A318, A319, A320, and A321 pertain to the Airbus A320 family. This should be taken into account while assigning the cockpit crews to the aircraft as each cockpit crew is eligible to work with a particular aircraft family. Moreover, some aircraft are not capable of performing specific flights where, for the instance, the number of passengers of particular fare exceeds the number of available seats.

It is noteworthy that the problem of fleet assignment concerns only the type of aircraft, but not the particular aircraft (Bazargan 2010). It was proved, that the complexity of the fleet assignment problem for three aircraft is NP-hard (Gu et al. 1994).

The next task is to assign an individual aircraft to each flight. This stage is called aircraft rotation or *aircraft routing*. If in the case of fleet assignment only particular types of aircraft were matched with flight legs, aircraft routing implies the assignment of an individual vehicle (Bazargan 2010). In other words, for each particular aircraft, it is necessary to determine the sequence of flight legs to be covered such that each leg is flown by exactly one aircraft. During this step, several requirements should be taken into account. One of them is the feasibility subject to obligatory maintenance check. According to Ben Ahmed, Zeghal Mansour, and Haouari (2018), an obligatory preventive maintenance check is performed periodically for all the aircraft before accumulating a defined quantity of flying hours since the last maintenance check. A feasible aircraft route in relation to maintenance check contains consistent flight legs. Each flight leg should be covered by one aircraft, and one aircraft cannot perform several flights at the same time. All the routes should satisfy the next conditions:

- The departure station of the first leg and the arrival station of the last leg must be the same maintenance station;
- The time passed between the arrival time of the last leg and the departure time of the first leg must be greater than the required maintenance check time;
- The total flying time should not exceed a specified time limit when the maintenance check must be done.

To undergo the maintenance check, aircraft must be landed for a certain amount of time at one of its maintenance bases, which is usually situated at the airline hubs (Bazargan 2010). The problem of aircraft routing is an NP-complete problem in general cases and has a polynomial size in the case of fixed fleet size (Parmentier 2013).

After the aforementioned steps, it is necessary to find the solution for the *crew pairing* problem. This stage implies the assignment of crews to each flight leg while fulfilling several complex work rules and minimizing the crew cost (Ben Ahmed, Zeghal Mansour, and Haouari 2018). As pilots are eligible to control an aircraft with a particular cockpit configuration, or in other words, they are qualified to steer only one aircraft family, the crew pairing problem is separately for each aircraft family (Shao, Sherali, and Haouari 2015). This task should also take into account matching the schedule of other aircraft crew members.

According to Ben Ahmed, Zeghal Mansour, and Haouari (2018), a *duty period* is a single workday of a crew that includes a sequence of flight legs with short rest periods, or sits, separating them. A *pairing* is a sequence of duty periods with overnight rests between consecutive periods. Each pairing begins and ends at the same station (the crew base). There are restrictions that the following concepts for the aircraft crew should be in the determined legal range:

- Layover duration between two consistent duties;
- Sit-time between consecutive flights.

Nevertheless, if two consecutive flight legs are performed by one aircraft, the sit-time between them can be reduced and be less than allowed minimum. In this case, it is equal to the aircraft turnaround time. There are also constraints regarding the following requirements:

- Maximum flying hours between two consecutive rests;
- Time in pairing away from the base;
- Duty duration;
- Maximum number of landings in one duty;
- Maximum number of duties in one pairing.

Moreover, after finishing a pairing, a crew should be provided with the rest time that is equal or exceeded the required minimum rest time.

To assign an aircraft and a crew to each flight, it is necessary to follow several requirements regarding aircraft maintenance and crew working rules:

- Each flight is covered by exactly one aircraft route and exactly one crew pairing;
- Each aircraft route and each crew pairing should be periodic and thereby repeats itself every day;
- Each aircraft route is maintenance feasible;
- The total number of required aircraft of each type should not exceed the available size of the corresponding sub fleet;

- Each pairing along with the corresponding duties should satisfy all those mentioned above constraints;
- The total number of required crews should not exceed the available number of crews.

The phase of crew pairing does not require the assignment of the individual crew members to crew pairings. However, this happens during the stage of *crew rostering*. It is noteworthy that the procedure of assigning the cockpit-crew and cabin-crew is not the same as the cockpit-crew is usually eligible to control the specific aircraft while cabin-crew can serve different fleet types (Bazargan 2010). Sometimes the problems of crew pairing and crew rostering are combined into crew scheduling problem. Nevertheless, we do not consider the problem of crew rostering in our study and for the sake of simplicity keep only crew pairing problem.

2.2 Delays

Following the information presented in Chapter 1.1, delays cause airline companies huge annual losses. Optimization in operation management influences the amount of delayed and canceled flights. Thus, according to the Bureau of Transportation Statistics (2019), all the situations of delayed flights in the US can be distinguished into five groups: Air Carrier Delay, National Aviation System Delay, Extreme Weather, Aircraft Arriving Late, and Security Delay. Definition of the groups:

- Air Carrier Delay: Circumstances within the control of the airline (e.g., maintenance or crew problems, aircraft cleaning, baggage loading, fueling, etc.)
- National Aviation System Delay: Issues attributable to the national aviation system that refers to a wide range of conditions, such as non-extreme weather conditions, airport operations, heavy traffic volume, and air traffic control.
- Extreme Weather: Weather conditions that are preventing airline operations from working in a regular way.
- Late-Arriving Aircraft: Delay of current departure caused by the later arrival of the previous flight, operated by the same aircraft.
- Security Delay: Delays or cancellations caused by evacuation, re-boarding of aircraft caused by security violation, broken screening equipment, or queues more than 29 minutes at security control areas.

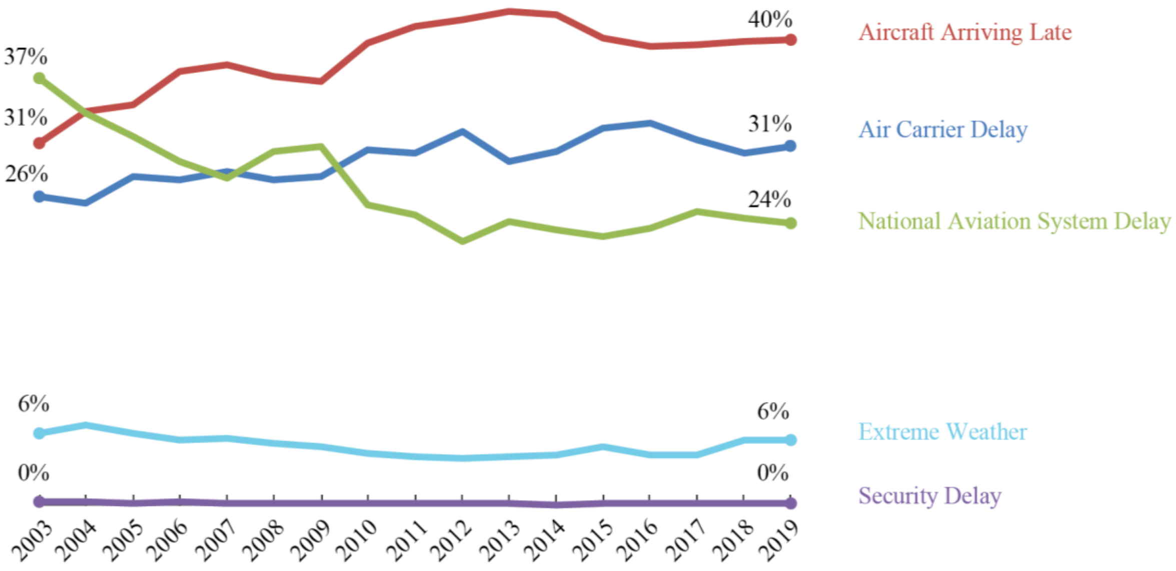

Figure 2 (Bureau of Transportation Statistics, 2019)

The most critical delay categories are Aircraft Arriving Late, Air Carrier Delay, and National Aviation System Delay (Figure 1). Some of the delays categorized as Aircraft Arriving Late or Air Carrier Delay can be avoided by optimization of airline planning processes. For example, by optimizing the schedules of crews and the assignment of aircraft, it is possible to prevent the downstream propagation of delays in the network of flights in the case of occurred disruption.

2.3 Robustness criteria

To avoid delays and cancellations, it is necessary to produce well-balanced schedules. This can be done by applying special techniques called robustness criteria that aim to increase the resistance of schedule to disruptions. Robustness criteria can be distinguished into two groups: *flexible* or *stable*. Flexibility means the fast capability to recover from an unpredictable delay while stability helps to avoid the situation of a delayed flight. Stable approaches require inserting or adjusting buffer times of flight legs (Ben Ahmed, Zeghal Mansour, and Haouari 2018). They are computed according to prior knowledge of delays', and inserted by slightly retiming flight legs. For this study, data about delays is not provided and therefore stable criteria cannot be applied. Besides, a stable approach makes the problem more complex since it is necessary to add decisions about retiming. Furthermore, retiming may not be efficient when schedules are tight, hence adjusting a flight departure time would make the aircraft connection infeasible.

In this study, the following flexible robustness criteria are taken: if the critical connection (when the connection time is larger than the legal minimum sit-time but less than the specified threshold value) is covered exclusively by an aircraft route or a crew pairing, the penalty is introduced. By doing this, additional idle time is promoted to absorb unpredictable disruptions, and the propagation of delays into downstream flights is mitigated. Another robust criterion is to force the crew to stay at the same aircraft to perform the next scheduled flight, assigned to this aircraft. In so doing, crews are less likely to be delayed because of short connection time. Robustness is promoted in our model by embedding the penalty terms into the objective function. These metrics penalize situations where the robustness criteria are not satisfied.

It is noteworthy, that the robustness criterion that prevents critical aircraft connections implicitly forces another criterion to take place. This criterion promotes a swap of aircraft within one route, which produces a more stable schedule (Burke et al. 2010), (Ionescu and Kliewer 2011). In this case, when one aircraft has a short connection time and another has a longer connection time and it overlaps the short one, the optimality can be reached by forwarding the second aircraft to the first route and the first aircraft to the second route (Figure 3).

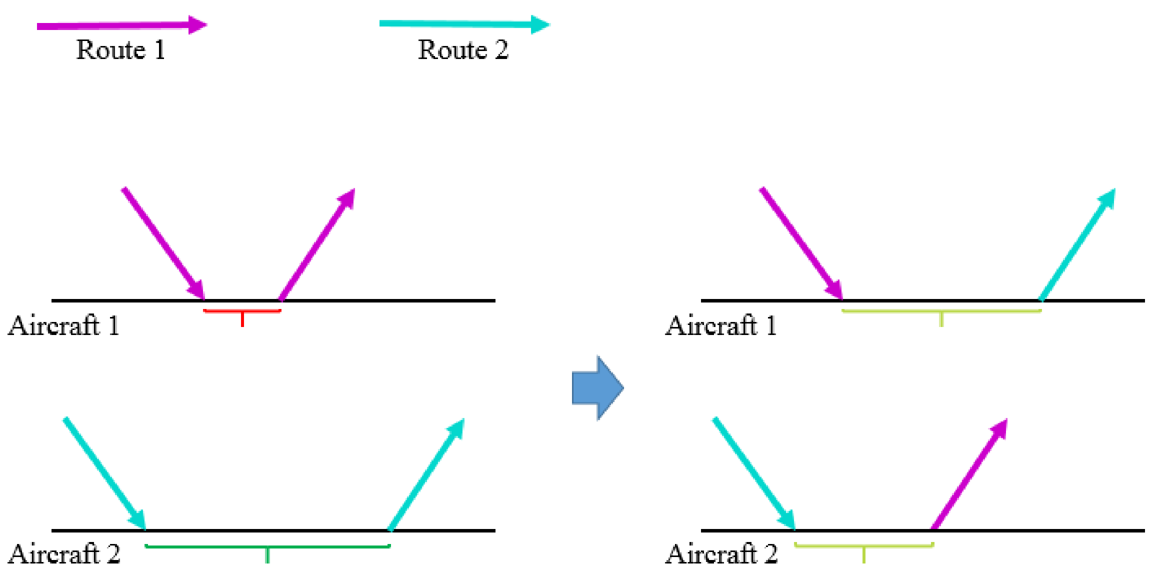

Figure 3 - Swap of flight legs

2.4 Conclusion

To solve the robust integrated fleet assignment, aircraft routing, and crew pairing problem, it is necessary to find a feasible set of fleet assignments, aircraft routes, and crew pairings that satisfies all the constraints and requirements regarding each of the problems and maximizes the profit while minimizing the chance of delay to occur. Profitability is represented by the objective function, which is the sum of the profit including a reward minus the sum of expenses including penalties from violated robust criteria, where:

- The profit consists of the estimated revenue from accommodated passengers while the reward is granted to each connection (whether it is critical or not) that is both covered by a crew pairing and an aircraft route. Hence, solutions, where the crew is following the aircraft, are promoted.
- Expenses are made up of fleet assignment cost while the penalty is imposed for each critical connection included in aircraft routes as well as critical connection covered by crew pairing without being covered by any aircraft route. In so doing, we avoid both aircraft and pairing connections that have a short buffer time.

3. Research questions & methodology

In this study, the following research questions are answered and the proposed methodology is aimed to help to create and to solve the mathematical model for the considered problems.

3.1 Research questions

1. In which sequence the model should be solved?

Historically, the problems of airline planning operations are solved in the next sequence: firstly fleet assignment, then aircraft routing, and afterward crew pairing. To avoid suboptimal solutions, those problems should be integrated into one model. However, it is possible to solve only relatively small instances using such a complex integrated model. To simplify the model, several mathematical techniques could be used, such as linearization, relaxation, and decomposition.

2. Which behavior the model is experience?

There are several ways to investigate the behavior of the model:

- Analyze the computational time needed to solve the model to optimality;
- Analyze the time of finding a feasible solution;
- See how large the optimality gap is after a fixed amount of running time;
- Examine whether it is reasonable to use the quickly-founded feasible solution or to spend more time on finding the optimal solution.

3. How costly are robustness measurements?

To introduce the robustness measurements in the model it is necessary to determine their influence on the objective function. Thus, we suggest using a quadratic penalty for the connections that are classified as short connections. To facilitate crews to follow an aircraft a reward is utilized. This reward is an independent parameter whose value is set empirically.

3.2 Methodology

In this study, there are four main leverages used to reach the desired results:

1) Mathematical modeling

The creation of the mathematical model was inspired by early published papers in this domain of study, while several of the implemented concepts are novel.

2) Linearization of model

To solve the created mathematical model it is necessary to use a commercial solver. As a non-linear model requires much more time to be solved, to speed up the process the model should be linearized. This goal can be reached using the reformulation-linearization technique (Sherali and Adams 1990, 1994) which implies variable substitution and transformation.

3) Programming implementation

To implement the model, an algebraic modeling language AMPL was used. This language is aimed to describe and solve large-scale optimization problems (Fourer, Gay, and Kernighan 2003).

4) Computational experiments

Computational experiments were carried out using a software package CPLEX, which is a mathematical solver aimed to solve linear, quadratic, and mixed-integer programming problems (IBM, n.d.).

5) Evaluation of results

To analyze the results it is necessary to evaluate the cost value of robustness measurements and behavior of the model.

4. Literature review

This chapter outlines the relevant airline planning research conducted in the past. They are distinguished into two sections: non-robust integrated models and robust integrated models.

4.1 Non-robust integrated models

In the literature review, due to the utility for the research, we focus on the integration of three operational stages: fleet assignment, aircraft routing, and crew pairing. Previously, because of the computational complexity, in most of the works, the integration of only two problems was considered. However, there are few, which propose and describe approaches to the integration of fleet assignment, aircraft routing, and crew pairing problems.

Eltoukhy, Chan, and Chung (2017) presented a survey with an overview of papers where the main airline problems are examined. The authors classified research into five different groups according to the covered problem: flight scheduling papers, fleet assignment papers, aircraft maintenance routing papers, crew scheduling papers (including crew rostering), and papers with the integrated models. For each group, the authors distinguished several subgroups according to the used solution methods as well as objective function and data. In conclusion, the advantages and the drawbacks of different solution approaches were discussed and the authors suggested which enhancement could be applied.

Sandhu and Klabjan (2007) are known to be the authors of the first published paper that investigates the idea of three-problem integration. To simplify the process of integration they disregarded maintenance requirement constraints. The authors proposed two separate methodologies to solve the integrated model. The first methodology is based on the Lagrangian relaxation and delayed column generation, while the second one utilizes Benders decomposition. It is worthy to note that Benders decomposition method is widely used in solving integrated models for airline planning as it is functioning well with mixed-integer programming problems. For example, it was used in research made by Papadakos (2009) and Shao, Sherali, and Haouari (2015), where fleet assignment, aircraft routing, and crew pairing problems integration was examined. Sandhu and Klabjan (2007) reached 3% of the average cost savings using the integrated approach in comparison with the sequential. Lagrangian relaxation seemed to be more efficient for the majority of instances. Besides, it should be noted that the authors performed the integration of fleet assignment and crew pairing using the enforced assignment of a pairing to the specific aircraft family.

Furthermore, Papadakos (2009) described another way of integrating the three aforementioned problems in one model and the computational experiments. Unlike Sandhu and Klabjan (2007), Papadakos (2009) considered maintenance requirements. In addition to his main integrational model, he proposed several alternative formulations. To reduce the number of constraints in the main model, the author uses Benders decomposition, where the crew pairing problem with short connections is decomposed into a column generation master problem and a subproblem. To accelerate the column generation, two heuristic methods are applied. The model is based on a crew-connection network and aircraft-connection network, where it is necessary to solve the shortest-path problem. For accelerating Benders decomposition, Papadakos (2009) used the improved version of Magnanti–Wong method which helps to compute a Pareto-optimal cut based on the Benders subproblem. The model was tested on seven sets of instances and solved to near-optimality.

Salazar-González (2014) proposed a heuristic approach to solve the integrated fleet assignment, aircraft routing, and crew pairing model based on an integer programming problem. He also separately drew attention to the crew rostering problem and its solution methods. The advantage of heuristic methods is the possibility to find a feasible solution to all the integrated problems while solving them sequentially can lead not only to a suboptimal solution but also to an infeasible one. In his work, Salazar-González (2014) used a similar representation of two directed graphs as in our research, where the first graph considers the aircraft routing problem and the second graph considers crew pairing problem. The mathematical formulation was tuned to meet the specific constraints of a regional carrier required for the whole problem. The heuristic method implies generating good crew solutions and then solving a mixed-integer linear programming problem.

Another relevant work was done by Shao, Sherali, and Haouari (2015). They presented an integrated model of fleet assignment, aircraft routing, and crew pairing problems, which also incorporates maintenance constraints, itinerary-based demands, and crew work requirements. Benders decomposition technique is used in this research along with the generation of Pareto-optimal cuts to speed-up the decomposition algorithm's convergence. The mathematical formulation of the problem in our work has similar representation as the model in the paper from Shao, Sherali, and Haouari (2015), as they used polynomially-sized node-arc flow network to describe the fleet assignment and aircraft routing problems, however, they did not utilize the resembling representation for crew pairing. Such an integrated approach brought for the authors an improvement in profit of an average of 8.4% as opposed to the sequential one.

4.2 Robust integrated models

To increase the resistance of airline planning models to delays and interruptions, the researchers started to implement the different robustness criteria to balance the produced schedules. To make a full overview of existing robust approaches, Agbokou (2004) presented a survey on the relevant optimization solutions. According to the survey, to deal with uncertainty, which occurs due to disruptions, two approaches are commonly used: the post-factum schedule re-optimization after a disruption occurs and the introduction of robustness during the planning. In the case of disruption, airlines are acting in a way to minimize the consequences by applying aircraft recovery, crew recovery, and passenger recovery models to reroute the resources. However, most of the research considers only aircraft recovery, as it is a more valuable resource from the company's point of view. A better decision could be to introduce uncertainty or incorporate the robustness during the planning stage.

Cordeau et al. (2001) did one of the first studies where a robustness criterion was implemented. Their criterion implied the crew to follow the aircraft if the connection time is too short, while the solution approach considered using Benders decomposition and column generation algorithms. This paper initiated other scientists who investigate airline planning to start using constraints, which help to build more delay-resistant schedules.

Four years later, Mercier, Cordeau, and Soumis (2005) presented a paper with enhanced flight connection restrictions. They introduced the possibility of limiting the number of short connections as well as forbidden the crew to change the aircraft during a short connection. The authors used the concept of *restricted connection* (a connection that is longer than the minimum short time but shorter than a certain threshold and this connection occurs between two flight legs that are not flown by the same aircraft) and imposed a penalty in the cases when such connections take place.

To produce a robust flight schedule several flexible approaches were historically utilized. Rosenberger, Johnson, and Nemhauser (2004) proposed an idea that flight schedules with short cycles (flight sequence with the same starting and ending airport) are less vulnerable to propagated flight cancellations. The authors also suggested to reduce hub connectivity (number of flight legs that are in a route that begins in one hub and ends in another hub with intermediate stops) as a disruption at one hub affects processes at another hub.

Another flexible robust technique was suggested by Smith and Johnson (2006) and concerns station purity. Station purity constrains the number of fleet types or crew compatible families that are used by the company at each airport to create more opportunities to swap aircraft of crews in the case of disruptions. However, this approach has a negative impact on computational efficiency and thus requires to apply more sophisticated solution methods.

Burke et al. (2010) presented a flexible criterion based on aircraft swap opportunities. A swap was determined as a reasonably long overlap between the ground times within two flight routes that allow a feasible ex-change of their aircraft. Swap opportunities are frequently used on the operation day to decrease the disruption influence by redistributing slack time between the aircraft rotations. Ionescu and Kliewer (2011) formulated crew pairing problem based on the set-partitioning and they shared a similar approach as Burke et al. (2010) in their research, but instead of swapping aircraft, Ionescu and Kliewer (2011) proposed swap opportunities within crew pairings while crew rostering remains feasible.

Tekiner, Birbil, and Bülbül (2009) defined crew pairing problem as a set-partitioning problem and examined one source of disruptions linked to additional flights that are inserted during operation (e.g. charter flights), that creates uncertainty during the schedule planning. The authors proposed some recovery operations using a robustness budget to avoid the delays or cancellations of settled flights while managing the additional flights.

Another study was done by Ben Ahmed, Zeghal Mansour, and Haouari (2018). To solve the problem of maintenance aircraft routing and crew pairing the authors suggested a robust approach based on aircraft routing and crew pairing graphs. The robustness criteria that were used promote the flight connections that are simultaneously covered by the aircraft and crew pairing, while the connections with too short buffer time are avoided. The model produced cost-efficient solutions with improved performance and reduced delay time.

Another type of robustness criteria, that is called stable criteria, are used to help to form the flight schedule before a delay occurs. Dück et al. (2012) distinguished delays into two groups: primary, that are cannot be controlled by airline operations management, and reactionary, that happen due to management instructions, such as waiting for the passengers from the late preliminary flight leg. The authors formulated an integrated stochastic model for crew scheduling and fleet assignment as a set-partitioning problem with reactionary delay propagation, caused by crews changing aircraft. To obtain a robust schedule, researchers used an indicator of stability based on the stochastic model.

Dunbar, Froyland, and Wu (2012) studied in their work the dependencies between aircraft routing and crew pairing. They investigated the influence of late-arriving aircraft or crew on the succeeding flights. The authors mentioned the importance of making the crew and aircraft routing decisions together for minimizing the cost-propagated delay. In later work, Dunbar, Froyland, and Wu (2014) enhanced their solution methodology using the information about the stochastic propagated delay.

Ben Ahmed et al. (2017) published a study considering aircraft routing and retiming using hybrid optimization-simulation methods. The goal of this work was to increase aircraft performance while decreasing the total delay as well as the number of delayed passengers. To this end, the authors presented a nonlinear mixed-integer programming model and suggested a Monte Carlo-based approach to regulate the departure times of aircraft. The researchers gained the improvement of the performance by 9.8–16.0%, while the cumulative delay was reduced by 25.4–33.1%, and the number of delayed passengers was reduced by 8.2–51.6% as opposed to the original airline solutions.

Cacchiani and Salazar-González (2017) presented two mixed-integer linear programming models that integrate three aforementioned airline planning problems: fleet assignment, aircraft routing, and crew pairing. They focused on minimizing a weighted sum of the number of aircraft routes, the number of crew pairings, and the waiting times of crews between consecutive flights with respect to the maintenance requirements. Cacchiani and Salazar-González (2017) have also applied robustness by reducing the necessity of crews to change the aircraft. The first model was called the “path-path” model as it introduces the crew pairings and the aircraft using path-based variables. In the second model, “arc-path”, the aircraft routes are indicated using arc-based variables and the crew pairings using path-based variables. For each of the presented models, the authors suggested separate exact algorithms with the corresponding names of “path-path” and “arc-path” methods. Both of them include three stages. Firstly, a lower bound is computed by the linear programming relaxation being solved to optimality using column generation on the path-based variables. Secondly, a heuristic solution (upper bound) is calculated using the variables generated in the first phase. In the third stage, the lower and upper bounds are used to compute an optimal solution. The “arc-path” method showed a better result and optimally solved all the instances that contained up to 172 flights of the regional carrier in the Canary Islands. Even if this research used exact methods to solve the integrated problem, the model was tuned to meet the requirements of a small regional carrier: perform maintenance at a single depot, flights are scheduled between 7 AM and 11 PM, eleven airports that are involved including four bases. Thus, the proposed solution cannot be used for a bigger airline company.

The recent work by Cacchiani and Salazar-González (2020) is focused on flight retiming along with the fleet assignment, aircraft routing, and crew pairing problem integration. The flight departure time is adjusted by choosing a better option from a set of discrete departure times. The authors reckon for maintenance requirements and crew working constraints. They also used robust criteria by penalizing too short and too long connection time, crew members changing aircraft within one connection, and a small penalty for the use of each aircraft. The authors suggested four two-phase heuristic algorithms based on a mixed-integer linear programming model using a similar approach from their earlier work in 2017, where path variables represented the crew pairings and arc variables represented the aircraft routes together with column-generation method applied to path variables. All four algorithms were tested on the instances of the regional air carrier and revealed their advantages and drawbacks, although one algorithm showed a better quality-complexity trade-off.

4.3 Conclusion

In this chapter, we presented an overview of published research focused on the integrated airline planning models and robust approaches. There are only a few papers where the fleet assignment, aircraft routing, and crew pairing problems are integrated, and no papers where robustness criteria are added to the integration of those problems. Thus, our study stands out from the previously made work. Some of the concepts used in this research, such as graph representation and robustness criteria, were inspired by the work of Ben Ahmed, Zeghal Mansour, and Haouari (2018). However, they integrated the models of only two problems.

In this work, a compact model is developed containing a polynomial number of constraints. This number depends on the number of flights and aircraft that create a polynomial function. Thus, we do not use heuristic and exact algorithms as, for example, set-partitioning where the exponential number of constraints used due to the generation of new variables.

5. Mathematical Formulation

5.1 Introduction

To fulfill the objective of the study – creation of a model that helps to derive a profitable flight schedule which is resistant to the propagation of delays and flight cancellations, it is crucial to implement using mathematical formulation an integration of three chosen airline problems and apply robustness criteria. To accomplish this, hereinafter we define terminology, notation, variables, and formulations needed to create a nonlinear mixed-integer programming model that is later transformed into a linear model. A similar mathematical formulation was used in the work made by Ben Ahmed, Zeghal Mansour, and Haouari (2018).

5.2 Problem notation

In this study, we refer to a set $L, j \in L$ as a set of daily flights to be executed by a group of aircraft families $f \in F$ that includes a set of aircraft types $k \in K^f$. The number of aircraft of type k is denoted by N_k . For each flight j the corresponding flying time is defined by t_j, the respective departure and arrival times are fixed and denoted as parameters T_j^D and T_j^A, departure stations and arrival stations are S_j^D and S_j^A, while S^k – a set of maintenance stations, $k \in K^f, f \in F$. In the following mathematical formulation, all the time parameters are expressed in minutes and therefore they are in the interval $[0,1440)$.

Aircraft routing graphs

For mathematical formulation, we associate the flight schedule with a digraph $G = (V, A)$ where each node $j \in V$ represents a flight leg and each arc $(i, j) \in A, A \equiv \cup A^k, f \in F, k \in K^f$ represents a feasible connection. An arc$(i, j) \in A^k$ if and only if an aircraft of type k can consecutively serve the flights pertaining to the to-node and the from-node of this arc. In addition, we denote respectively the set of arcs that are incident to, and outgoing from, node $j \in V$ by δ_j^- and δ_j^+ (Figure 4).

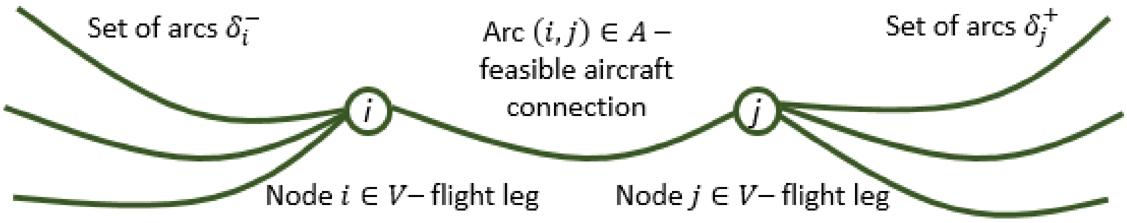

Figure 4

It is also necessary to define four arc subsets A_1^k, A_2^k, A_3^k, and A_4^k, that are included in the set of arcs A^k, for each $f \in F, k \in K^f$. They are described as follows:

- An arc $(i,j) \in A_1^k$ if and only if a maintenance check can be done between the arrival of the flight i and the departure of flight j, and both flights are performed consecutively on the same day. Thus, $(i,j) \in A_1^k \leftrightarrow$
$$\begin{cases} S_i^A \equiv S_j^D \\ S_i^A \in S^k \\ T_i^A + T_M \leq T_j^D \end{cases}$$
where T_M is the time needed to perform the maintenance check.
- An arc $(i,j) \in A_2^k$ if and only if a maintenance check can be done between the arrival of the flight i and the departure of flight j, and the same aircraft is covering flight leg j the day after serving flight leg i. Thus, $(i,j) \in A_2^k \leftrightarrow$
$$\begin{cases} S_i^A \equiv S_j^D \\ S_i^A \in S^k \\ T_j^D < T_i^A + T_M \leq T_j^D + 1440 \end{cases}$$
- An arc $(i,j) \in A_3^k$ if and only if a maintenance check cannot be done between the arrival of the flight i and the departure of flight j, and both flights are performed consecutively on the same day. Thus, $(i,j) \in A_3^k \leftrightarrow$
$$\begin{cases} S_i^A \equiv S_j^D \\ S_i^A \notin S^k or\, T_j^D < T_i^A + T_M \\ T_i^A + T_T \leq T_j^D \end{cases}$$
where T_T is a turnaround time, or the time needed for the aircraft to be ready to perform the next flight.
- An arc $(i,j) \in A_4^k$ if and only if a maintenance check cannot be done between the arrival of the flight i and the departure of flight j, and the same aircraft is required to serve flight leg j the day after serving flight leg i. Thus, $(i,j) \in A_4^k \leftrightarrow$

$$\begin{cases} S_i^A \equiv S_j^D \\ S_i^A \notin S^k \ or \ T_j^D + 1440 < T_i^A + T_M \\ T_j^D < T_i^A + T_T \leq T_j^D + 1440 \end{cases}$$

The set of maintenance arcs is denoted by $A_M^k \equiv A_1^k \cup A_2^k$ for each type of aircraft $f \in F, k \in K^f$, and $A_{NM}^k \equiv A^k \setminus A_M^k$ is the set of non-maintenance arcs. It is noteworthy, that the arcs that belong to $A_1^k \cup A_3^k$ represent the connections between a pair of consecutive flights that depart on the same day, while the connections from the union $A_2^k \cup A_4^k$ represent a wraparound ground connection between a pair of flights flown on two consecutive days. If the day of the departure of a flight does not correspond to the day of the arrival of the same flight, this flight is considered as a wraparound flight and persists to the subset $L^{WAF} \subset L$.

Furthermore, to introduce itinerary-based flight demands, where an itinerary is a planned route for a passenger, the defined below notation is used:

- Π – set of itineraries, where $\Pi_j \subset \Pi$ – the subset of itineraries that include flight $j, j \in L$.
- H – set of all fare classes.
- C_{hk} – passenger seat capacity for fare class $h \in H$ on aircraft of type $k \in K$.
- $\bar{\pi}_{ph}$- mean demand for fare class $h \in H$ on flight $j \in L$ within itinerary $p \in \Pi^j$.
- r_{ph}- estimated revenue from one ticket for fare class $h \in H$ on flight $j \in L$ within itinerary $p \in \Pi^j$.

Besides, in the objective function, we use the same fleet assignment cost representation, that was suggested by Zeghal Mansour et al. (2011):

$$c_{jk} = \bar{c}_{jk} + \sum_{h \in H} o_{jh} \left(\sum_{p \in \Pi_j} \bar{\pi}_{ph} - C_{hk} \right)^+ \tag{1}$$

where $\bar{c}_{jk}$ – the fixed cost of assigning an aircraft of fleet type k to flight leg j, o_{jh}– the opportunity cost per spilled passenger on flight leg j, and $(.)^+ \equiv \max\{0, .\}$. The concept of spilled passengers occurs when the expected demand for fare class h exceeds the capacity of the assigned aircraft. Hence, such representation of cost includes fixed operating charges and the opportunity cost of spilled passengers. Opportunity cost per spilled passenger is calculated in the following way:

$$o_{jh} = 0.2 \sum_{p \in \Pi_j} r_{ph} \bar{\pi}_{ph} / \sum_{p \in \Pi_j} \bar{\pi}_{ph}, \quad \forall j \in L, h \in H$$

Airline companies usually estimate the number of spilled passengers to be around 20%. The remaining surplus passengers are either rebooked or upgraded to a higher fare, so no losses are incurred (Shao, Sherali, and Haouari 2015). The value of spilled passengers is represented as a contribution of each itinerary to flight $j \in L$.

Crew pairing graph

Let $L_D \subset L$ denote the set of flights that depart from the base station, and $L_A \subset L$ the set of flights that arrive at the base station. The sit-time between consecutive flights i and j that are included in the same duty period is denoted as T_{ij}^{ST}, and T_{ij}^{LO} is the layover time between two consecutive flights i and j that belong to two consecutive duty periods within the same pairing. Hereinafter, we use the following notation:

- $T_{min}^{ST}/T_{max}^{ST}$ – minimum/maximum crew sit-time between two consecutive flights within the same duty;
- T_{max}^{DF} – maximum duty flight duration;
- T_{max}^{DD} – maximum duty duration, assuming that the longest flight duration is shorter than the maximum duty duration;
- $T_{min}^{LO}/T_{max}^{LO}$ – minimum/maximum layover duration between two consecutive duties within the same pairing. It is noteworthy, that T_{min}^{LO} is also the minimum rest time after completing a pairing;
- $T_{min}^{DP}/T_{max}^{DP}$ – minimum/maximum pairing duration;
- N_{max}^{L} – maximum number of landings within a duty;
- N_{max}^{D} – maximum number of duties within a pairing.

We define a crew pairing graph as $G^{CP} = (\bar{V}, B)$ where to obtain a set of nodes $\bar{V}$ a dummy start node is added to V, $\bar{V} \equiv V \cup \{0\}$, so node 0 represents both the start and the end of a pairing and each node $j \in V$ represents a flight leg. Each arc $(i, j) \in B, B \equiv \cup B^f,\ f \in F$ represents a feasible connection. An arc $(i, j) \in B^f$ if and only if a crew that is eligible to an aircraft of family f can consecutively serve the flights pertaining to the to-node and the from-node of this arc. In addition, we denote respectively the set of arcs that are incident to, and outgoing from, node $j \in \bar{V}$ by $\bar{\delta}_j^-$ and $\bar{\delta}_j^+$ (Figure 5).

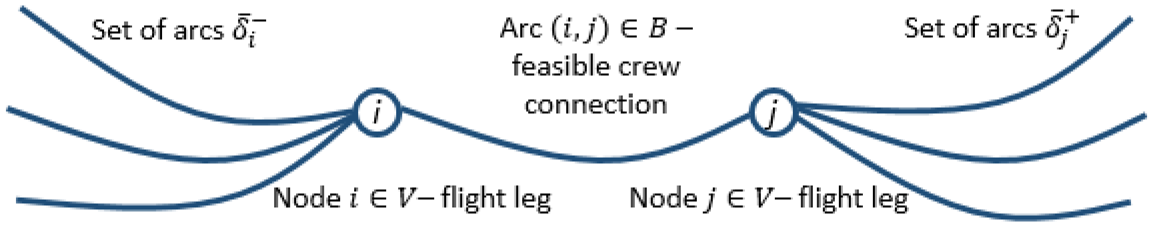

Figure 5

An arc $(i, j) \in B^f$ if and only if the next conditions are true:

- the arrival station S_i^A of flight i coincides with the departure station S_j^D of flight j;
- the total connection time is greater than or equal to the minimum sit-time and smaller than the maximum layover duration;
- the same crew that consecutively serves flights i and j is eligible to aircraft family $f \in F$.

Moreover, for each aircraft family $f \in F$, for each flight node $j \in L_D$ that departs from the crew base corresponds an arc $(0, j) \in B_D^f$, where B_D^f is a subset of departing arcs, and for each flight node $j \in L_A$ that arrives at the crew base corresponds an arc $(j, 0) \in B_A^f$, where B_A^f is a subset of arrival arcs.

It is also necessary to define two arc subsets B_1^f and B_2^f, that are included in the set of arcs B^f, for each $f \in F$. They are described as follows:

1. An arc $(i, j) \in B_1^f$ if and only if legs i and j can be consecutively served by the same crew during the same duty period. That is, each arc $(i, j) \in B_1^f$ corresponds to a short rest period within a duty. Therefore, $(i, j) \in B_1^f \leftrightarrow$
   i. The sit-time $T_{ij,f}^{ST}$ is bounded by aircraft turnaround time and the maximum allowed sit-time, i.e. $T_{ij,f}^{ST} \in [T_{T,f}, T_{max}^{ST}]$;
   ii. The maximum flying time with a duty is satisfied, i.e. $t_i + t_j \leq T_{max}^{DF}$;
   iii. The maximum duty duration is satisfied, i.e. $t_i + t_j + T_{ij,f}^{ST} \leq T_{max}^{DD}$, where the sit-time is defined as follows:

- $T_{ij,f}^{ST} = T_j^D - T_i^A$, if $T_j^D > T_i^A$, and means that the arrival of flight i and the departure of flight j occur on the same day;
- $T_{ij,f}^{ST} = T_j^D + 1440 - T_i^A$, if $T_j^D < T_i^A$, and means that the departure of flight j occurs on the next day than the arrival of flight i.

2. An arc $(i,j) \in B_2^f$ if and only if legs i and j can be consecutively served by the same crew in two consecutive duty periods within the same pairing. Hence, each arc $(i,j) \in B_2^f$ corresponds to a layover within a multi-day pairing, where the layover time $T_{ij,f}^{LO}$ is computed as follows:
   i. The layover duration is bounded by the minimum and the maximum layover time $T_{ij,f}^{LO} \in [T_{min}^{LO}, T_{max}^{LO}]$;
   ii. The maximum duration of the pairing is satisfied, i.e. $T_{min}^{DP} \leq t_i + t_j + T_{ij,f}^{LO} \leq T_{max}^{DP}$, where the layover time is defined as follows:
      - $T_{ij,f}^{LO} = T_j^D - T_i^A$, if $T_i^A + T_{min}^{LO} \leq T_j^D \leq T_i^A + T_{max}^{LO}$, and means that arrival of flight i and the departure of flight j occur on the same day;
      - $T_{ij,f}^{LO} = T_j^D + 1440 - T_i^A$, if $T_i^A + T_{min}^{LO} \leq T_j^D + 1440 \leq T_i^A + T_{max}^{LO}$, and means that the departure of flight j occurs on the next day after the arrival of flight i;
      - $T_{ij,f}^{LO} = T_j^D + 2880 - T_i^A$, if $T_i^A + T_{min}^{LO} \leq T_j^D + 2880 \leq T_i^A + T_{max}^{LO}$, and means that the departure of flight j occurs two days after the arrival of flight i;

Therefore, for a connection to be pertained to B_2^f arc it is necessary for layover and pairing duration to be within the legal range.

To be able to track the connections that violate robustness criteria we introduce a set of short connections $B^S \subset B_1$ that include the connections with sit-time shorter than the default minimum sit-time for the aircraft T_{min}^{AST}, but longer than the legal minimum sit-time for crew.

Decision variables

x_{ij} – binary variable that takes 1 if arc $(i,j) \in A$ is selected, and 0 otherwise.

u_{jk}– total accumulated flying hours for aircraft of type $f \in F, k \in K^f$ since its last maintenance check after serving flight leg $j \in L$.

w_{jk} – binary variable that equals 1 if flight leg $j \in L$ is assigned to an aircraft of type $f \in F, k \in K^f$, and 0 otherwise.

N_{ph}^{PAS} – number of passengers flying within fare class $h \in H$ and itinerary $p \in \Pi$.

y_{ij}– binary variable that takes 1 if arc $(i, j) \in B$ is selected, and 0 otherwise.

z_{ij} – binary variable that takes 1 if a crew follows an aircraft on a connection $(i, j) \in A \cap B$.

N_{jf}^{L} – total number of landing for a crew that is eligible to an aircraft family $f \in F$ after serving flight leg $j \in L$.

T_{jf}^{DF} – total accumulated duty flight duration for a crew that is eligible for an aircraft family $f \in F$ after serving flight leg $j \in L$.

T_{jf}^{DD} – total accumulated duty duration for a crew that is eligible to an aircraft family $f \in F$ after serving flight leg $j \in L$.

N_{jf}^{D} – total accumulated number of duties for a crew that is eligible for an aircraft family $f \in F$ after serving flight leg $j \in L$.

T_{jf}^{DP} – total accumulated duration of pairing for a crew that is eligible to an aircraft family $f \in F$ after serving flight leg $j \in L$.

d_{jf} – integer variable that corresponds to the duration (in days) of the crew pairing that is eligible to an aircraft family $f \in F$, that ends with flight j (if any), $j \in L$.

5.3 A compact nonlinear mixed-integer programming model

Critical connections

For each arc $(i, j) \in A^k$, where $f \in F, k \in K^f$, the aircraft planned idle time I_{ij}^{a} is defined as the difference between the (i, j)-connection time and aircraft turnaround time. Thus,

$$I_{ij}^{a} = \begin{cases} T_j^D - T_i^A - T_T, & \text{if } T_j^D \geq T_i^A + T_T \\ T_j^D + 1440 - T_i^A - T_T, & \text{otherwise} \end{cases} \quad \forall f \in F, k \in K^f, (i, j) \in A^k \tag{2}$$

To avoid critical connections that more likely lead to delays and hence stimulate robust schedules to be generated, the quadratic penalty q_{ij}^{a} for the aircraft connections (i, j) with short buffer time is introduced. This penalty is used for each aircraft connection with idle time shorter than a preset aircraft connection cushion time I^{a}. The penalty is computed as follows:

$$q_{ij}^{a} = \begin{cases} \left(I^{a} - I_{ij}^{a}\right)^{2}, & \text{if } I_{ij}^{a} < I^{a} \\ 0, & \text{otherwise} \end{cases} \qquad \forall f \in F, k \in K^{f}, (i,j) \in A^{k} \tag{3}$$

To apply the same principle to the crew connection $(i,j) \in B_1$, when a connection is considered as short and the sit-time is within the legal range, the planned crew idle time is defined as follows:

$$I_{ij}^{c} = \begin{cases} T_j^D - T_i^A - T_{min}^{ST}, & \text{if } T_j^D \geq T_i^A + T_{min}^{ST} \\ T_j^D + 1440 - T_i^A - T_{min}^{ST}, & \text{otherwise} \end{cases} \qquad \forall (i,j) \in B_1 \tag{4}$$

Thus, using a preset crew connection cushion time I^c, the set of critical connections is defined as follows:

$$B^C = \{(i,j) \in B_1 \backslash B^S : I_{ij}^c < I^c\}$$

If a crew does not follow the aircraft during a critical connection, i.e. a critical connection is covered by a crew pairing but not an aircraft route, then a penalty q_{ij}^c is applying. The penalty is computed as follows:

$$q_{ij}^{c} = \left(I^{c} - I_{ij}^{c}\right)^{2}, \qquad \forall (i,j) \in B^C \tag{5}$$

For a better understanding of crew connections classification, refer to Figure 6.

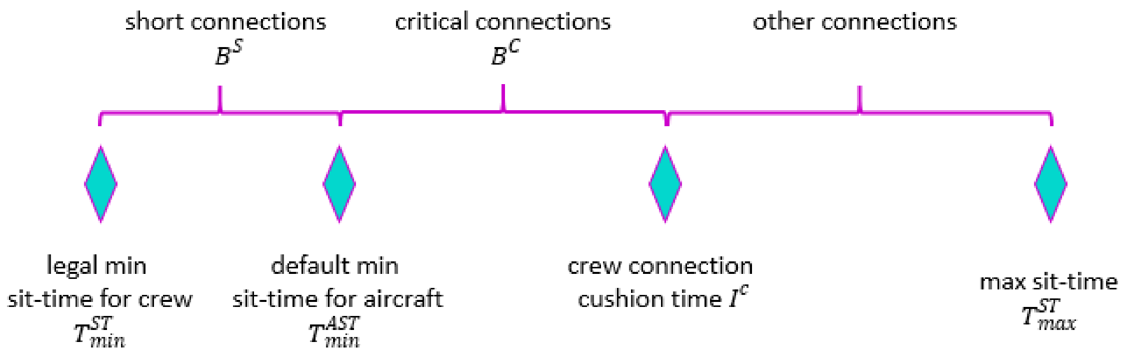

Figure 6 - Crew connections classification

Objective function

The objective function (6) is aimed to maximize the sum which consists of the profit that is represented by the demand multiplied by the estimated revenue and the reward R of the robust connection when the connection is covered both by crew pairing and aircraft. The reward R is a parameter and its value is set empirically. It is noteworthy, that on the short connection a crew follows an aircraft by default which is implied by constraints (39). The objective function also includes losses due to fleet assignment costs and penalties for critical connection in aircraft route as well as for critical connection covered by the crew without being included in any aircraft route.

$$Maximize \sum_{p\in\Pi}\sum_{h\in H} r_{ph}N_{ph}^{PAS} - \sum_{j\in L}\sum_{f\in F}\sum_{k\in K^f} c_{jk}w_{jk} + \sum_{(i,j)\in B\setminus B^S} Rz_{ij} - \sum_{(i,j)\in A} q_{ij}^{a}x_{ij} - \sum_{(i,j)\in B^C} q_{ij}^{c}\left(y_{ij} - z_{ij}\right) \tag{6}$$

Aircraft route feasibility

$$\sum_{f\in F, k\in K^f} w_{jk} = 1, \qquad \forall j \in L \tag{7}$$

$$\sum_{(i,j)\in\delta_j^-\cap A^k} x_{ij} = w_{jk}\,, \qquad \forall j \in L, f \in F, k \in K^f \tag{8}$$

$$\sum_{(j,i)\in\delta_j^+\cap A^k} x_{ji} = w_{jk}\,, \qquad \forall j \in L, f \in F, k \in K^f \tag{9}$$

Constraints (7) ensure that each flight leg is covered by exactly one aircraft. Constraints (8) and (9) require each flight to possess exactly one predecessor and one successor and that both of them are assigned to the same aircraft type. Thus, as both w and x are binary, those constraints ensure that the solution consists of cycles or cyclic rotations covered and each rotation covers a set of flights using a particular type of aircraft.

$$u_j^k x_{ij} = t_j x_{ij}, \qquad \forall j \in L, f \in F, k \in K^f, (i,j) \in \delta_j^- \cap A_M^k \tag{10}$$

$$u_j^k x_{ij} = (u_i^k + t_j)x_{ij}, \qquad \forall j \in L, f \in F, k \in K^f, (i,j) \in \delta_j^- \cap A_{NM}^k \tag{11}$$

$$t_j \le u_j^k \le T_{max}^M, \qquad \forall j \in L, f \in F, k \in K^f \tag{12}$$

For each type of aircraft, constraints (10) represent that the accumulated time since the last maintenance check is equal to the duration of the previous flight in the case if the maintenance check was performed right before this flight. In other cases, to update the accumulated time since the last maintenance check it is necessary to add the duration of the previous flight to the last recorded value as it is shown in constraints (11). Constraints (12) prevent the situation of exceeding defined quantity of flying hours since the last maintenance check and parameter T_{max}^{M} indicates the maximum number of flying hours that aircraft can perform without the maintenance check.

$$\sum_{(i,j)\in A_M^2 \cup A_M^4} x_{ij} \leq N_k - \sum_{j\in L^{WAF}} w_{jk}, \quad \forall f \in F, k \in K^f \tag{13}$$

As the nature of rotations is cyclic and at the beginning of each day the number of available aircraft should exclude wraparound flight aircraft from the previous day, constraints (13) ensure that the total number of aircraft in service does not exceed the fleet size.

Itinerary feasibility

$$\sum_{h\in H} \sum_{p\in \Pi_j} N_{ph}^{PAS} \leq \sum_{h\in H} \sum_{f\in F} \sum_{k\in K^f} C_{hk}\, w_{jk}, \qquad \forall j \in L \tag{14}$$

$$0 \leq N_{ph}^{PAS} \leq \bar{\pi}_{ph}, \qquad \forall p \in \Pi, h \in H \tag{15}$$

Constraints (14) guarantee that the total number of passengers traveling on the flight does not exceed the available seat capacity of the assigned aircraft. At the same time, to avoid the situation of overbooking, constraints (15) make sure that the total number of passengers traveling on any particular itinerary for each fare class is not more than the total expected demand.

Duty feasibility

$$\sum_{(i,j)\in \bar{\delta}_j^- \cap B^f} y_{ij} = \sum_{k\in K^f} w_{jk}, \qquad \forall j \in L, f \in F \tag{16}$$

$$\sum_{(j,i)\in \bar{\delta}_j^+ \cap B^f} y_{ji} = \sum_{k\in K^f} w_{jk}, \qquad \forall j \in L, f \in F \tag{17}$$

Constraints (16) and (17) force crews to be assigned to the flights covered by aircraft that crews are eligible to be assigned to.

$$\sum_{(j,i)\in B_D^f} y_{ji} = \sum_{(i,j)\in B_A^f} y_{ij}\,, \qquad \forall j \in L, f \in F \tag{18}$$

Constraints (18) require the number of start arcs to be equal to the number of end arcs. These constraints help to create the cyclic crew pairings.

$$N_{jf}^L y_{ij} = y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap \left(B_2^f \cup B_D^f\right) \tag{19}$$

$$N_{jf}^L y_{ij} = \left(N_{if}^L + 1\right) y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap B_1^f \tag{20}$$

$$1 \le N_{jf}^L \le N_{max,f}^L, \qquad \forall j \in L, f \in F \tag{21}$$

Constraints (19) and (20) help to count the number of landings, as well as constraints (21) that verify that the maximum amount of landings within one duty is not exceeded.

$$T_{jf}^{DF} y_{ij} = t_j y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap \left(B_2^f \cup B_D^f\right) \tag{22}$$

$$T_{jf}^{DF} y_{ij} = \left(T_{if}^{DF} + t_j\right) y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap B_1^f \tag{23}$$

$$t_j \le T_{jf}^{DF} \le T_{max,f}^{DF}, \qquad \forall j \in L, f \in F \tag{24}$$

Constraints (22) and (23) track the total flying time and constraints (24) put the restriction on the total duration of flights within a duty.

$$T_{jf}^{DD} y_{ij} = t_j y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap \left(B_2^f \cup B_D^f\right) \tag{25}$$

$$T_{jf}^{DD} y_{ij} = \left(T_{if}^{DD} + T_{ijf}^{ST} + t_j\right) y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap B_1^f \tag{26}$$

$$t_j \le T_{jf}^{DD} \le T_{max,f}^{DD}, \qquad \forall j \in L, f \in F \tag{27}$$

Constraints (25) and (26) compute the duty duration time while constraints (27) help to avoid the exceeding of maximum duty duration time.

Pairing feasibility

$$N_{jf}^{D} y_{ij} = y_{ij}, \qquad \forall j \in L_D, f \in F, (i,j) \in \bar{\delta}_j^- \cap B_D^f \tag{28}$$

$$N_{jf}^{D} y_{ij} = N_{if}^{D} y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap B_1^f \tag{29}$$

$$N_{jf}^{D} y_{ij} = (N_{if}^{D}+1) y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap B_2^f \tag{30}$$

$$1 \leq N_{jf}^{D} \leq N_{max,f}^{D}, \qquad \forall j \in L, f \in F \tag{31}$$

Constraints (28)-(30) count the number of duties within each pairing while constraints (31) ensure that the restrictions for the maximum number of duties within one pairing are satisfied.

$$T_{jf}^{DP} y_{ij} = t_j y_{ij}, \qquad \forall j \in L_D, f \in F, (i,j) \in \bar{\delta}_j^- \cap B_D^f \tag{32}$$

$$T_{jf}^{DP} y_{ij} = \left(T_{if}^{DP} + T_{ijf}^{ST} + t_j\right) y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap B_1^f \tag{33}$$

$$T_{jf}^{DP} y_{ij} = \left(T_{if}^{DP} + T_{ijf}^{LO} + t_j\right) y_{ij}, \qquad \forall j \in L, f \in F, (i,j) \in \bar{\delta}_j^- \cap B_2^f \tag{34}$$

$$T_{min}^{DP} y_{j0} \leq T_{jf}^{DP}, \qquad \forall j \in L_A, f \in F, (j,0) \in \bar{\delta}_j^+ \cap B_A^f \tag{35}$$

$$t_j \leq T_{jf}^{DP} \leq T_{max,f}^{DP}, \qquad \forall j \in L, f \in F \tag{36}$$

Constraints (32)-(34) help to track a pairing duration time, or, in other words, time away from the base. Constraints (35) set the restrictions on the minimum duration of pairing while constraints (36) limit the maximum duration of pairing.

Number of available crews

Constraints (37), (38), and (43) ensure that the total number of crews in service does not exceed the number of available crews qualified for the required aircraft families. If a pairing ends with flight j and therefore the duration of pairing is $T_j^{DP} y_{j0}$, then after adding a compulsory post-pairing rest time the total duration of pairing is $\left(T_j^{DP} y_{j0} + T_{min}^{LO}\right)$ minutes. Thus, the total duration of pairing in days is $d_j = \left\lceil \frac{T_j^{DP} y_{j0} + T_{min}^{LO}}{1440} \right\rceil$ days. Since all flights repeat themselves every day, then one pairing requires d_j crews.

$$T_{j,f}^{DP} y_{j0} + T_{min}^{LO} \leq 1440\, d_{jf}, \qquad \forall j \in L_A, f \in F, (j,0) \in \bar{\delta}_j^{+} \cap B_A^f \tag{37}$$

$$\sum_{(j,0) \in B_A^f} d_{jf} \leq N_f^{crew}, \qquad \forall f \in F \tag{38}$$

where N_f^{crew} is the number of available crews qualified for the required aircraft families.

Short and critical connections

$$y_{ij} \leq x_{ij}, \qquad \forall (i,j) \in B^S \cap A \tag{39}$$

$$0 \leq z_{ij} \leq x_{ij}, \qquad \forall (i,j) \in B \cap A \tag{40}$$

$$0 \leq z_{ij} \leq y_{ij}, \qquad \forall (i,j) \in B \cap A \tag{41}$$

Constraints (39) promote crew to follow aircraft on the short connection while constraints (40) and (41) in the case if $z_{ij} = 1$ enforce crew to follow aircraft on the other connections $(i,j) \in B$.

Integrity and non-negativity constraints

$$(x, y, z, w) \text{ binary} \geq 0 \tag{42}$$

$$(u, d, N^{PAS}, N^{L}, T^{DF}, T^{DD}, N^{D}, T^{DP}) \text{ integer} \geq 0 \tag{43}$$

Remark 1. According to Sherali, Bae, and Haouari (2010), it is possible to replace constraints (14) with the following inequality:

$$\sum_{h \in H} \sum_{p \in \Pi_j} N_{ph}^{PAS} \leq \sum_{f \in F} \sum_{k \in K^f} \sum_{h \in H} \tilde{C}_{jkh} \, w_{jk}, \qquad \forall j \in L \tag{44}$$

where $\tilde{C}_{jkh} \equiv \min\left\{C_{hk}, \sum_{p \in \Pi_j} \bar{\pi}_{ph}\right\}, \ \forall j \in L, h \in H, f \in F, k \in K^f$

It is possible to replace constraints (15) with:

$$0 \leq N_{ph}^{PAS} \leq \tilde{\pi} \equiv \min\left\{\bar{\pi}_{ph}, \max_{k \in K^f, f \in F} C_{hk}\right\}, \qquad \forall p \in \Pi, h \in H \tag{45}$$

Remark 2.

It is possible to eliminate variable w from the formulation using the constraints (8) and (9). In the constraints (7) by replacing w with the left-hand side of either (8) or (9) we obtain:

$$\sum_{(i,j) \in \delta_j^- \cap A^k} x_{ij} = 1, \forall j \in L \tag{46}$$

$$\sum_{(j,i) \in \delta_j^+ \cap A^k} x_{ji} = 1, \forall j \in L \tag{47}$$

Moreover, constraints (8) and (9) can be replaced with:

$$\sum_{(i,j) \in \delta_j^- \cap A^k} x_{ij} = \sum_{(j,i) \in \delta_j^+ \cap A^k} x_{ji}, \qquad \forall j \in L, f \in F, k \in K^f \tag{48}$$

Following the same logic, we can eliminate w-variable from constraints (16) and (17). The left-hand side of those constraints can be used as well for replacing w in constraints (7). Thus, we obtain:

$$\sum_{(i,j)\in\bar{\delta}_j^- \cap B^f} y_{ij} = 1, \qquad \forall j \in L, f \in F \tag{49}$$

$$\sum_{(j,i)\in\bar{\delta}_j^+ \cap B^f} y_{ji} = 1, \qquad \forall j \in L, f \in F \tag{50}$$

$$\sum_{(i,j)\in\bar{\delta}_j^- \cap B^f} y_{ij} = \sum_{(j,i)\in\bar{\delta}_j^+ \cap B^f} y_{ji}, \qquad \forall j \in L, f \in F \tag{51}$$

However, as w-variable is used in the linearization process, we decide to keep in the mathematical formulation.

Remark 3.

Usually, airline companies do not calculate pairing costs as a monetary value, since crew members have a fixed monthly payment that does not depend on the assignments to flights. Instead, it is relevant to evaluate the cost of overnight stays of crews in locations different than a crew base. This principle is widely used by European airlines (Haouari et al, 2019). Therefore, companies can use a metric called "flight-time credit" (FTC), which is a difference between the duration of pairing and the pairing flying time (Ben Ahmed, Zeghal Mansour, and Haouari 2018). Thus, FTC is equal to the sum of sit-times and layovers. In this study, pairing costs are ignored in the objective function because these costs are negligible when compared to the assignment cost. However, FTC is tracked as a separate value in order to compare the quality of crew assignments.

Remark 4.

To examine the influence of robustness techniques on the model, it is necessary to present a non-robust integrated model. The non-robust objective function is the following:

$$Maximize \sum_{p\in\Pi}\sum_{h\in H} r_{ph}N_{ph}^{PAS} - \sum_{j\in L}\sum_{f\in F}\sum_{k\in K^f} c_{jk}w_{jk} \qquad (52)$$

The rest of the constraints (7)-(39) remain the same except for the removal of the constraints (40) and (41) that enforce crew to follow aircraft on the non-short connections.

In Chapter 6 the results of running the non-robust integrated model are compared to the results from the robust one.

5.4 Model Linearization

By paying attention to the constraints (10), (11), (19), (20), (25), (26), (28)-(30), (32)-(34), (37) we can see that they are non-linear due to multiplication between integer and binary variables which yields the non-linearity of the whole model. This increases the complexity of solving the problem.

Thus, to improve the solvability of the model we apply the reformulation-linearization technique presented by Sherali and Adams (1990, 1994) in order to obtain a linear representation of our model. Linearization is performed by defining new nonnegative artificial variables for each existing cross-product term and redefining the constraints where substituted variables are used. The full linear equivalent model with the notation can be found in Appendix A.

Aircraft routing linearization

Using the following transformation from Shao, Sherali, and Haouari (2015) and the similar linearization process as in Ben Ahmed, Zeghal Mansour, and Haouari (2018), we substitute constraints (10) and (11) with a new artificial variables α and $\bar{\alpha}$:

$$\alpha_{ij}^k = u_i^k x_{ij}, \qquad \forall f \in F, k \in K^f, (i,j) \in A^k \qquad (A.1)$$

$$\bar{\alpha}_{ij}^k = u_j^k x_{ij}, \qquad \forall f \in F, k \in K^f, (i,j) \in A^k \qquad (A.2)$$

Thus, linearized constraints (10) and (11) become:

$$\bar{\alpha}_{ij}^{k} = t_j x_{ij}, \qquad \forall f \in F, k \in K^f, j \in L_k, (i,j) \in A_M^k \tag{A.3}$$

$$\bar{\alpha}_{ij}^{k} = \alpha_{ij}^{k} + t_j x_{ij}, \qquad \forall f \in F, k \in K^f, j \in L_k, (i,j) \in A_{NM}^k \tag{A.4}$$

Next step is to multiply constraints (12) by x_{ij} and $x_{ji}, \forall (i,j) \in A, \forall (j,i) \in A$. Using the substitution from constraints (A.1) and (A.2), we obtain:

$$t_j x_{ij} \leq \bar{\alpha}_{ij}^{k} \leq T_{max,f}^{M} x_{ij}, \qquad \forall f \in F, k \in K^f, (i,j) \in A^k \tag{A.5}$$

$$t_i x_{ij} \leq \alpha_{ij}^{k} \leq T_{max,f}^{M} x_{ij}, \qquad \forall f \in F, k \in K^f, (i,j) \in A^k \tag{A.6}$$

In addition, we multiply constraints (46) and (47) by u_j^k. After rearranging indices, we obtain:

$$\sum_{(i,j) \in A^k} \bar{\alpha}_{ij}^{k} = u_j^k, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{A.7}$$

$$\sum_{(j,i) \in A^k} \alpha_{ji}^{k} = u_j^k, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{A.8}$$

Hence, modifying constraints (A.5) using (A.7), obtain:

$$t_j \sum_{(i,j) \in A^k} x_{ij} \leq u_j^k \leq T_{max,f}^{M} \sum_{(i,j) \in A^k} x_{ij}\,, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{A.9}$$

Proposition 1. Constraints (A.3)-(A.9) can be substituted with the equivalent constraints (A.10)-(A.12):

$$\sum_{(j,i)\in A^k} \alpha_{ji}^k = t_j + \sum_{(i,j)\in A_{NM}^k} \alpha_{ij}^k, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{A.10}$$

$$t_j x_{ji} \le \alpha_{ji}^k \le \left(T_{max,f}^M - t_i\right) x_{ji}, \qquad \forall f \in F, k \in K^f, (j,i) \in A_{NM}^k \tag{A.11}$$

$$t_j x_{ji} \le \alpha_{ji}^k \le T_{max,f}^M x_{ji}, \qquad \forall f \in F, k \in K^f, (j,i) \in A_M^k \tag{A.12}$$

Proof. From the equalities (A.7) and (A.8) obtain:

$$\sum_{(i,j)\in A^k} \bar{\alpha}_{ij}^k = \sum_{(j,i)\in A^k} \alpha_{ji}^k, \qquad \forall f \in F, k \in K^f, j \in L_k$$

Which is equivalent to:

$$\sum_{(i,j)\in A_M^k} \bar{\alpha}_{ij}^k + \sum_{(i,j)\in A_{NM}^k} \bar{\alpha}_{ij}^k = \sum_{(j,i)\in A^k} a_{ji}^k, \qquad \forall f \in F, k \in K^f, j \in L_k$$

Using the substitution from constraints (A.3) and (A.4), equality transforms into the following form along with the elimination of $\bar{\alpha}$-variables:

$$\sum_{(j,i)\in A^k} a_{ji}^k = t_j + \sum_{(i,j)\in A_{NM}^k} a_{ij}^k, \qquad \forall f \in F, k \in K^f, j \in L_k$$

In addition, by substituting constraints (A.4) into (A.5) for $\forall (i,j) \in A_{NM}^k, f \in F, k \in K^f$, we get constraints (A.11) and by keeping (A.6) for $\forall (i,j) \in A_M^k, f \in F, k \in K^f$ we obtain (A.12). ■

For convenience, we introduce a new parameter:

$$b_j^t = \begin{cases} T_{max,f}^M - t_j, & \forall f \in F, k \in K^f, (i,j) \in A_{NM}^k \\ T_{max,f}^M, & \forall f \in F, k \in K^f, (i,j) \in A_M^k \end{cases}$$

Thus, using the new parameter and after reorganizing indices, constraints (A.11) and (A.12) can be transformed into the following form:

$$t_i x_{ij} \leq a_{ij}^k \leq b_{ij}^t x_{ij}, \qquad \forall f \in F, k \in K^f, (i,j) \in A^k \tag{A.13}$$

Crew pairing linearization

Now, using a similar process and following Ben Ahmed, Zeghal Mansour, and Haouari (2018) we linearize constraints (19) and (20) utilizing artificial variables β and $\bar{\beta}$ within the next substitution:

$$\beta_{ij}^f = N_{i,f}^L y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.1}$$

$$\bar{\beta}_{ij}^f = N_{j,f}^L y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.2}$$

Hence, equalities (19) and (20) become:

$$\bar{\beta}_{ij}^f = y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \cup B_D^f \tag{B.3}$$

$$\bar{\beta}_{ij}^f = \beta_{ij}^f + y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \tag{B.4}$$

Next, multiplying constraints (21) with $y_{ji}, \forall f \in F, (j,i) \in B^f \backslash B_D^f$, we get:

$$y_{ji} \leq N_{j,f}^L y_{ji} \leq N_{max,f}^L y_{ji}, \qquad \forall f \in F, (j,i) \in B^f \backslash B_D^f$$

Using the equality (B.2), after reorganizing indices the inequality becomes:

$$y_{ij} \leq \beta_{ij}^f \leq N_{max,f}^L y_{ij}, \qquad \forall f \in F, (j,i) \in B^f \backslash B_D^f \tag{B.5}$$

Similarly, multiplying constraints (21) with $y_{ij}, \forall f \in F, (i,j) \in B^f \backslash B_A^f$, we get:

$$y_{ij} \leq \bar{\beta}_{ij}^{f} \leq N_{max,f}^{L} y_{ij}, \qquad \forall f \in F, (i,j) \in B^{f} \backslash B_{A}^{f} \tag{B.6}$$

In addition, we multiply constraints (49) and (50) by the respective $N_{j,f}^{L}$ and linearizing, obtain:

$$\sum_{(i,j)\in B^{f}} \bar{\beta}_{ij}^{f} = N_{j,f}^{L}, \qquad \forall f \in F, k \in K^{f}, j \in L_{k} \tag{B.7}$$

$$\sum_{(j,i)\in B^{f}} \beta_{ji}^{f} = N_{j,f}^{L}, \qquad \forall f \in F, k \in K^{f}, j \in L_{k} \tag{B.8}$$

Proposition 2. Constraints (B.3)-(B.8) can be substituted with the equivalent constraints that together with constraints (49) and (50) are still valid in continuous relaxation sense:

$$\sum_{(j,i)\in B^{f}} \beta_{ji}^{f} = 1 + \sum_{(i,j)\in B_{1}^{f}} \beta_{ij}^{f}, \qquad \forall f \in F, k \in K^{f}, j \in L_{k} \tag{B.9}$$

$$y_{ij} \leq \beta_{ij}^{f} \leq \left(N_{max,f}^{L} - 1\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_{1}^{f} \tag{B.10}$$

$$y_{ij} \leq \beta_{ij}^{f} \leq N_{max,f}^{L} y_{ij}, \qquad \forall f \in F, (i,j) \in B_{2}^{f} \cup B_{A}^{f} \tag{B.11}$$

Proof. From the equalities (B.7) and (B.8) obtain:

$$\sum_{(i,j)\in B^{f}} \bar{\beta}_{ij}^{f} = \sum_{(j,i)\in B^{f}} \beta_{ji}^{f}, \qquad \forall f \in F, k \in K^{f}, j \in L_{k}$$

Which is equivalent to:

$$\sum_{(i,j)\in B_1^f} \bar{\beta}_{ij}^f + \sum_{(i,j)\in B_2^f} \bar{\beta}_{ij}^f + \sum_{(i,j)\in B_D^f} \bar{\beta}_{ij}^f = \sum_{(j,i)\in B^f} \beta_{ji}^f, \qquad \forall f \in F, k \in K^f, j \in L_k$$

Using the substitution from constraints (B.3) and (B.4), equality transforms into the following form along with the elimination of $\bar{\beta}$-variables:

$$\sum_{(i,j)\in B_1^f} \left(\beta_{ij}^f + y_{ij}\right) + \sum_{(i,j)\in B_2^f} y_{ij} + \sum_{(i,j)\in B_D^f} y_{ij} = \sum_{(j,i)\in B^f} \beta_{ji}^f, \qquad \forall f \in F, k \in K^f, j \in L_k$$

Thus, together with constraints (49) and (50), it is equivalent to:

$$1 + \sum_{(i,j)\in B_1^f} \beta_{ij}^f = \sum_{(j,i)\in B^f} \beta_{ji}^f, \qquad \forall f \in F, k \in K^f, j \in L_k$$

■

Using the same substitution principle, it is possible to linearize constraints (22), (23), (25), (26), (28)-(30), (32)-(34), (37). The substitution should be defined in the following way:

$$\gamma_{ij}^f = T_i^{DF} y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.12}$$

$$\bar{\gamma}_{ij}^f = T_j^{DF} y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.13}$$

$$\mu_{ij}^f = T_i^{DD} y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.14}$$

$$\bar{\mu}_{ij}^f = T_j^{DD} y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.15}$$

$$\varphi_{ij}^f = N_i^D y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.16}$$

$$\bar{\varphi}_{ij}^{f} = N_j^D y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.17}$$

$$\omega_{ij}^{f} = T_i^{DP} y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.18}$$

$$\bar{\omega}_{ij}^{f} = T_j^{DP} y_{ij}, \qquad \forall f \in F, (i,j) \in B^f \tag{B.19}$$

Hence, the aforementioned constraints are rewritten as follows:

$$\sum_{(j,i)\in B^f} \gamma_{ji}^{f} = t_j + \sum_{(i,j)\in B_1^f} \gamma_{ij}, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{B.20}$$

$$t_i y_{ij} \le \gamma_{ij}^{f} \le \left(T_{max,f}^{DF} - t_j\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \tag{B.21}$$

$$t_i y_{ij} \le \gamma_{ij}^{f} \le T_{max,f}^{DF} y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \cup B_A^f \tag{B.22}$$

$$\sum_{(j,i)\in B^f} \mu_{ji}^{f} = \sum_{(i,j)\in B_1^f} T_{ij,f}^{ST}\, y_{ij} + t_j + \sum_{(i,j)\in B_1^f} \mu_{ij}^{f}, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{B.23}$$

$$t_i y_{ij} \le \mu_{ij}^{f} \le \left(T_{max,f}^{DD} - T_{ij,f}^{ST} - t_j\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \tag{B.24}$$

$$t_i y_{ij} \le \mu_{ij}^{f} \le T_{max,f}^{DD} y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \cup B_A^f \tag{B.25}$$

$$\sum_{(j,i)\in B^f} \varphi_{ji}^{f} = \sum_{(i,j)\in B_D^f \cup B_2^f} y_{ij} + \sum_{(i,j)\in B_1^f \cup B_2^f} \varphi_{ij}^{f}, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{B.26}$$

$$y_{ij} \le \varphi_{ij}^{f} \le (N_{max,f}^{D} - 1) y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \tag{B.27}$$

$$y_{ij} \leq \varphi_{ij}^{f} \leq N_{max,f}^{D} y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \cup B_A^f \tag{B.28}$$

$$\sum_{(j,i)\in B^f} \omega_{ji}^{f} = \sum_{(i,j)\in B_1^f} T_{ij,f}^{ST}\, y_{ij} + \sum_{(i,j)\in B_2} T_{ij,f}^{LO} y_{ij} + t_j + \sum_{(i,j)\in B_1^f \cup B_2^f} \omega_{ij}^{f}, \tag{B.29}$$

$$\forall f \in F, k \in K^f, j \in L_k$$

$$t_i y_{ij} \leq \omega_{ij}^{f} \leq \left(T_{max,f}^{DP} - T_{ij,f}^{ST} - t_j\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \tag{B.30}$$

$$t_i y_{ij} \leq \omega_{ij}^{f} \leq \left(T_{max,f}^{DP} - T_{ij,f}^{LO} - t_j\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \tag{B.31}$$

$$\omega_{j0}^{f} + T_{min,f}^{LO} \leq 1440 d_{j,f}, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{B.32}$$

6. Computational study

In this chapter, the description of test instances provided by United Airlines Company and based on the historical data is presented as well as computational experiments and their results. Similar data instances were used earlier in the research made by Shao, Sherali, and Haouari (2015). All tests were performed on an Intel Core i7-8700 CPU, 3.2GHz processor computer with 16GB of RAM, and the model was implemented using AMPL modeling language and CPLEX 12.8 solver with default settings. AMPL code for the robust integrated model can be found in Appendix B.

6.1 Data description

Data used for running the mathematical model were provided by United Airlines Company. They are 4 instances: HS1, HS2, HS3, HS4, where each of them includes several text files that contain information about the aircraft families, fleet content, flights, and itineraries. In order to transform the company data into a readable form for the commercial solver, the python programming language was used to generate the data files. Python code for the data conversion can be found in Appendix C.

Each instance contains three fare classes: Business, Economy Plus, and Economy. Company's fleet comprises three aircraft families and five aircraft types:

- Airbus 320
    - Airbus 319
    - Airbus 320
- Boeing 757
    - Boeing 752
    - Boeing 763
- Boeing 777
    - Boeing 772

In Table 2 the number of flights defined for each data instance is shown.

Table 2

Instance	Number of flights
HS1	128
HS2	154
HS3	246
HS4	354

Data files with the information about aircraft families comprise the following data: minimum sit-time for aircraft $T_{min}^{ACST} = 45$ minutes, minimum sit-time for crew eligible for this aircraft $T_{min}^{ST} = 30$ minutes, maximum sit-time for crew $T_{max}^{ST} = 240$ minutes (4 hours), minimum layover time $T_{min}^{LO} = 480$ minutes (8 hours). The values of the number of landings N_{max}^{L}, duration of pairing T_{max}^{DP}, and the number of duties within one pairing N_{max}^{D} that vary depending on the aircraft family are shown in Table 3.

Table 3

Instance	Aircraft	N_{max}^{L}	T_{max}^{DP}	N_{max}^{D}
HS1	B757	6	3240	3
	B777	4	4320	3
	A320	6	3240	3
HS2	B757	4	2880	3
	B777	4	4320	3
	A320	4	2880	3
HS3	B757	4	2880	3
	B777	4	4320	3
	A320	4	2880	3
HS4	B757	4	2880	3
	B777	4	4320	3
	A320	4	2880	3

Maintenance stations for each aircraft family are presented below:

- B757: BOS, DEN, IAD, JFK, LAS, LAX, LGA, MCO, ORD, PDX, SAN, SEA, SFO;
- B777: DEN, EZE, GIG, GRU, IAD, LAX, LHR, ORD, SEA, SFO, TPE;
- A320: BOS, DCA, DEN, IAD, LAS, LAX, LGA, MCO, MEX, MSP, ORD, PDX, SAN, SEA, SFO, SNA;

Data files with the information about fleet include the next information from Table 4, Table 5, and Table 6:

Table 4

Family	Type	T_{max}^{M}	T_M	T_T	Hourly cost
A320	A319	2700	420	38	5800
	A320	2700	420	40	5900
B757	B752	3300	390	45	6400
	B763	3300	390	45	6400
B777	B772	3900	360	70	9800

Table 4 contains the information about aircraft family, aircraft type, maximum flying time before the maintenance, time needed to perform the maintenance, turnaround time, and hourly cost of aircraft utilization.

Table 5

		Business	Economy Plus	Economy
A320	A319	8	40	72
	A320	12	36	90
B757	B752	16	45	108
	B763	25	60	110
B777	B772	36	89	223

The number of seats disposed of each fare class for each aircraft type is shown in Table 5.

Table 6 – Number of aircraft

	A320		B757		B777	
	A319	A320	B752	B763	B772	Total
HS1	10	7	14	3	1	35
HS2	12	20	16	3	2	53
HS3	18	31	26	5	2	82
HS4	26	45	37	7	3	118

Table 6 comprises the number of available aircraft of each type for each data instance.

Data files with the information about the schedule of daily flights include the following information: flight ID, departure station, departure time, arrival station, arrival time, and the duration of the flight.

Each line in data files with the information about itineraries includes itinerary ID, fare class, price of fare class, mean demand, and flight IDs covered by this itinerary.

Crew pairing restrictions that are not included in the provided data files: maximum duty flight duration $\mathrm{T}_{\mathrm{max}}^{\mathrm{DF}} = 480$ minutes (8 hours), maximum duty duration $\mathrm{T}_{\mathrm{max}}^{\mathrm{DD}} = 720$ minutes

(12 hours), minimum duration of pairing $\mathrm{T}_{\min}^{\mathrm{DP}} = 300$ minutes (5 hours), and maximum layover duration $\mathrm{T}_{\max}^{\mathrm{LO}} = 1980$ minutes (33 hours). Besides, aircraft cushion time and crew cushion time are set to be $I^a = 60$ minutes and $I^c = 60$ minutes respectively, and reward for crew following the aircraft is $R = 10000$. Additionally, as the number of crews that the airline company possesses is not provided, the used estimated number is 1.5 times more than the number of flights. The crew bases are chosen to be located at the following airports: ORD, IAH, LAX, EWR, SFO, IAD, DEN, and CLE.

6.2 Computational experiments

6.2.1 Computational experiments on United Airlines data

After running the model using the data provided by United Airlines Company it was discovered that the complexity of the model and the structure of data instances does not allow to obtain feasible solutions. Two families of the constraints that were influencing the model the most are the constraints regarding the number of duties within one pairing and the constraints regarding the total flight duration within one duty. As the maximum flight duration within one duty was selected to be not more than 480 minutes (8 hours), it was necessary to modify the long-haul flights to match these constraints. As the removal of those flights and their outbound would influence the flight-network structure, it was decided to reduce the flying time of flights that exceed 400 minutes. Thus, the modifications were applied to 4 flights in the data instance HS1 and 21 flights in the data instance HS4, while no flights were modified in the instances HS2 and HS3. The maximum duration of the flight in the instance HS1 was 470 min, while in HS4 it was 517 min.

Another obstacle that arises is the fact that after performing 3 duties (which is the maximum number of duties within one pairing) a crew is more likely to finish not at the base station, which triggers the infeasibility as each crew is required to finish their pairing at the base station. This happens because the developed model is more suitable for the point-to-point network structure, while United Airlines Company uses the hub-and-spoke route system. In the pure point-to-point system passengers do not need to have a transfer in the intermediate airport as they travel directly to their destinations. In hub-and-spoke structure passengers need to have a transfer at the hub to reach their destination, except the situation when the hub is the origin or the destination itself (Cook and Goodwin 2008). Thus, due to the data instances structure, it was decided to relax these constraints.

Table 7 - Robust model

	Duality gap	Best bound	Objective function	# of short crew conn.	# of crit. crew conn.	# of crit. aircraft conn.	# of conn. where crew follows the aircraft	FTC	Passengers revenue	Cost of aircraft assignment	Reward from crew following the aircraft	Penalty for crit. aircraft conn.	Penalty for crit. conn. w/o crew following aircraft	Total solve system time	Total solve user time
HS1	Opt	-192023	-192023	0	1	47	3	57481	1452270	1618720	30000	53524	2048	4433	167706
HS2	2.29%	1097178	1072580	0	0	9	5	71093	2898260	1874130	50000	1555	0	9334	360791

Table 8 - Non-robust model

	Duality gap	Best bound	Objective function	# of short crew conn.	# of crit. crew conn.	# of crit. aircraft conn.	# of conn. where crew follows the aircraft	FTC	Passengers revenue	Cost of aircraft assignment	Reward from crew following the aircraft	Penalty for crit. aircraft conn.	Penalty for crit. conn. w/o crew following aircraft	Total solve system time	Total solve user time
HS1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
HS2	Opt	1069220	1069220	0	3	91	Missing	67960	2898260	1829040	NA	NA	NA	1976	46499

Table 7 shows the results obtained after running the model on the provided data instances. The running time was selected to be within 18000 minutes (5 hours). As the number of flights in the instances HS3 and HS4 is very big, it was impossible to obtain a feasible solution for them. Thus, only the results for the instances HS1 and HS2 are shown. To evaluate the contribution of robustness the results for the non-robust model are shown as well in Table 8. Due to the structure of the model, it is impossible to track the number of connections where a crew follows an aircraft for the non-robust model.

For the instance HS1, it was possible to find a robust solution with 1 critical crew connection and 47 critical aircraft connections, while having 3 connections where crew follows the aircraft. The results for this instance comprise quite a high penalty for critical aircraft connections (53524). For this instance, the cost of aircraft assignment exceeds the passenger revenue. However, it takes twice less time to find the solution for the data instance HS1 than for the instance HS2 while the number of flights in HS1 is 128 and 154 in HS2. It is noteworthy that the objective function is a relative value which is influenced by the robustness reward and penalties. It is used as a leverage to enforce robustness to be applied to the model and thus cannot display the real profitability. Therefore, we cannot compare the value of the objective functions of robust and non-robust models. Besides, it appeared that finding the feasible solution for the instance HS1 using the non-robust model requires more time than the selected time limit.

Comparing the results for the instance HS2 we can see, that the difference in the aircraft assignment cost of the robust and non-robust model is 45,090 which is a relatively small number, while the number of critical crew connections for the robust model is reduced by 100% in comparison with non-robust one (3 critical crew connections for non-robust and 0 critical crew connections for robust), and the number of critical aircraft connections is reduced by 90% (91 against 9), which is also a good indicator. However, the solution of the robust model takes much more CPU time than the solution of a non-robust one (9,334 system CPU seconds for robust model against 1,976 system CPU seconds for non-robust, and 360,791 user CPU seconds for robust model against 46,499 user CPU seconds for non-robust). We can also outline, that the flight-time credit (FTC) increased for 4.6% for the robust solution (67,960 for non-robust model against 71,093 for robust one).

6.2.2 Computational experiments on artificial data

To examine the model more thoroughly, it was decided to create several artificial data instances based on the obtained data from United Airlines, but with a smaller number of flights. Thus, three instances A1, A2, and A3 were created based on the real instances HS1, HS2, and HS3 respectively. New artificial instances contain the following number of flights (Table 9), while the other parameters correspond to the parameters of the original instances HS1, HS2, and HS3:

Table 9

Instance	Number of flights
A1	90
A2	68
A3	94

The results of running the robust integrated model on the artificial instances are shown in Table 10. The running time was kept to be within 18000 minutes (5 hours). For evaluation of the influence of the robustness techniques, the results for the non-robust model are presented in Table 11.

Comparing the results from Table 10 and Table 11, we can see the significant decrease in the number of critical connections both for crews and aircraft in the case of robust planning (from 67% up to 100% decrease). Information about the percentage change in costs of a robust solution in comparison with the non-robust one is shown in Table 12. We can see, that the cost of aircraft assignment increased by 0.1% - 4.5%, while the difference between the FTC value of the non-robust model and the robust model varies from -11% to 13%.

Thus, the results show the efficiency of the robust techniques due to the decreased number of critical connections with non-significant aircraft assignment cost increase. The results also demonstrate the absence of a tendency regarding the increase or decrease of FTC cost while using the robust model. This means that the crew pairing cost can both increase and decrease while applying robustness techniques. Besides, the relatively small value of FTC in comparison with aircraft assignment cost (up to 3.9% of aircraft assignment cost) indicates its negligibility.

Table 10 - Robust model

	Duality gap	Best bound	Objective function	# of short crew conn.	# of crit. crew conn.	# of crit. aircraft conn.	# of conn. where crew follows the aircraft	FTC	Passengers revenue	Cost of aircraft assignment	Reward from crew following the aircraft	Penalty for crit. aircraft conn.	Penalty for crit. conn. w/o crew following aircraft	Total solve system time	Total solve user time
A1	Opt	-93824.8	-93824.8	0	0	8	1	36131	1146480	1248410	10000	1894	0	18	1218
A2	Opt	514336	514336	0	0	1	5	31825	1321020	856580	50000	100	0	22	1659
A3	Opt	3936820	3936820	0	0	1	2	47949	5582020	1664980	20000	225	0	21	2372

Table 11 - Non-robust model

	Duality gap	Best bound	Objective function	# of short crew conn.	# of crit. crew conn.	# of crit. aircraft conn.	# of conn. where crew follows the aircraft	FTC	Passengers revenue	Cost of aircraft assignment	Reward from crew following the aircraft	Penalty for crit. aircraft conn.	Penalty for crit. conn. w/o crew following aircraft	Total solve system time	Total solve user time
A1	Opt	-95558.3	-95558.3	0	6	24	NA	40636	1146480	1242040	NA	NA	NA	27	1254
A2	Opt	501358	501358	0	1	13	NA	32164	1321020	819658	NA	NA	NA	15	245
A3	Opt	3918470	3918470	0	5	11	NA	42386	5582020	1663550	NA	NA	NA	23	2382

Table 12 – Cost changes of the robust solution in comparison with the non-robust one

Instance	Aircraft assignment cost change, %	FTC change, %
A1	+0.5%	-11%
A2	+4.5%	-1%
A3	+0.1%	+13%

In Table 13 time parameters are presented and include the time needed to find a feasible solution, the optimal solution, and total calculation time for both robust and non-robust models which are run on each artificial data instance. The following notation is used:

- O – the time needed to solve the model to optimality (in seconds);
- F – the time of finding a feasible solution (in seconds).

Table 13

	Robust model			Non-robust model		
	F	O	Total (root+branch&cut)	F	O	Total (root+branch&cut)
A1	76	149	164 sec	66	219	220 sec
A2	93	309	309 sec	50	50	50 sec
A3	174	333	336 sec	126	302	303 sec

Table 14

A1					
Robust model			Non-robust model		
Time	Duality gap	Best bound	Time	Duality gap	Best bound
76 sec	7.54%	-92495.8	66 sec	4.17%	-94549.6
84 sec	2.79%	-92556.9	119 sec	3.91%	-94790.5
99 sec	0.86%	-93103.5	209 sec	0.33%	-95469.5
A2					
Robust model			Non-robust model		
Time	Duality gap	Best bound	Time	Duality gap	Best bound
93 sec	6.76%	522383.1	50 sec	0.00%	501358
131 sec	3.48%	522297.3			
308 sec	0.74%	518143.3			
A3					
Robust model			Non-robust model		
Time	Duality gap	Best bound	Time	Duality gap	Best bound
174 sec	1.01%	3947216.4	126 sec	0.54%	3920962.6
291 sec	0.54%	3945620.1	188 sec	0.09%	3920763.1
300 sec	0.02%	3937780.1	295 sec	0.02%	3919281.5

In Table 14 the information about the duality gap and the best bound at different moments of the model run is provided. For the data instance A1 a feasible solution with a big duality gap was found relatively fast for both robust and non-robust models. In the case of the robust model, the solution was quickly improved to have a duality gap of 0.86%, but afterwards, it took a lot of time to reach the optimality. For the non-robust model, it took a greater amount of time to reach a small duality gap and finally the optimal solution. The same performance experienced the robust model tested on the instance A2, where a feasible solution with a big duality gap was found quickly but took a lot of time to reduce the gap. However, the non-robust model was solved fast and directly to optimality. In the case of A3, it took time to reach a feasible solution, but once it was reached, the duality gap appeared to be very small and thus it would be more reasonable to stop the model at this moment than to wait approximately twice more time to find the optimal solution. All artificial data instances were solved within a smaller computational time in comparison with the original instances which indicates proportionality of computational time and problem size.

6.3 Analysis of the results

The computational results show the possibility of using the model where three airline problems are integrated into one with applied robustness techniques. We can notice, that with the increase in the number of flights the processor execution time increases as well. For the artificial instances, the computational time needed to find the optimal solution with the non-robust model proportionally depends on the number of flights, in contrast with the robust model. Therefore, in the case of the robust model, the computational times depends not only on the problem size but also on the applied robustness techniques. The time of finding the first feasible solution and the rate of duality gap decrease is varying from instance to instance, and no dependencies connected to model size are found. However, the time needed to find the first feasible and the optimal solution is bigger for the robust model, than for the non-robust.

After testing the model on real and artificial data with the different number of flights, we can state that due to the complexity of the model it is preferable to use it for regional carriers whose number of daily flights relatively small and whose network structure is the point-to-point structure.

It was also discovered, that to enforce the crew to follow the aircraft on the connections, the reward should be selected enough big to be able to influence the objective function. It is shown as well, that the choice of the quadratic penalty for critical aircraft and crew connections meets the expectations and helps to reduce the number of critical connections. However, the application of the robustness increases the computational time and the aircraft assignment cost, but not significantly. Thus, it proves the relevance of including the robustness features within the model. To examine how the robust model carries into effect the cost savings when disruptions occur in the network system and how it affects the fleet assignment cost, it is reasonable to use a simulation tool in the further research.

7. Conclusion and future work

7.1 Conclusion

The goal of this study was to create the model which integrates three airline planning problems: fleet assignment, aircraft routing, and crew pairing, and which increases the disruption resistance of operational schedule due to the robustness techniques. After that, the developed model was tested on the data instances provided by United Airlines Company. Computational results show the influence of the problem size and network structure on the efficiency of the model.

Using the robustness techniques reduces the number of critical connections and forces crew to follow the aircraft on the connections, but it increases the aircraft assignment cost. Running the model on the hub-and-spoke network structure affects the model efficiency as it is more suitable for the point-to-point system. With increasing problem size, the computational time is increasing as well.

7.2 Future work

In future work, to simplify and accelerate the solution process, relaxation and decomposition methods could be applied to the model. Relaxation of the linear programming problem is a model obtained by omitting integer and binary constraints on variables of the initial problem. Hereupon, all the variables become continuous except the one which remains integer and which indicates if a flight leg is assigned to an aircraft of a specific type or not. As a result, we obtain a mix-integer programming problem with only one integer variable.

The next step is solving the mix-integer programming problem using a commercial solver. By doing this, the solution for the fleet assignment problem is obtained as well as a lower bound which is the value of the objective function of the relaxed problem. Subsequently, the value of lower bound will be used for the comparison with the value of the objective function of final results. Thereby, the problem is hence decomposed into several subproblems of aircraft routing and crew pairing, where the number of subproblems is depending on the number of aircraft families. Therefore, it is required to solve the resulting problem for each aircraft family.

In this study, the developed robust integrated model was tested on the data instances with the hub-and-spoke network structure. To examine the behavior of the model more precisely, we suggest to test it on the data instances with the point-to-point network structure.

Besides, it is also possible to examine the robustness contribution to the airline planning process with disruptions using a simulation tool.

Reference list

Agbokou, Claudine Biova. 2004. “Robust Airline Schedule Planning: Review and Development of Optimization Approaches.” Thesis, Massachusetts Institute of Technology.

Air Transport Action Group. “Facts & Figures”. 2018a. Available from https://www.atag.org/facts-figures.html.

Ball, Michael, Cynthia Barnhart, Martin Dresner, Mark Hansen, Kevin Neels, Amedeo Odoni, Everett Peterson, Lance Sherry, Antonio Trani, Bo Zou. 2010. “Total Delay Impact Study: A Comprehensive Assessment of the Costs and Impacts of Flight Delay in the United States.” Final report, National Center of Excellence for Aviation Operations Research (NEXTOR). Available from https://isr.umd.edu/NEXTOR/pubs/TDI_Report_Final_10_18_10_V3.pdf

Bazargan, Massoud. 2010. Airline Operations and Scheduling. 2nd ed. Surrey: Ashgate Publishing Limited.

Ben Ahmed, Mohamed, Wisal Ghroubi, Mohamed Haouari, and Hanif D. Sherali. 2017. “A hybrid optimization simulation approach for robust weekly aircraft routing and retiming.” *Transportation Research Part C: Emerging Technologies* 84: 1–20.

Ben Ahmed, Mohamed, Farah Zeghal Mansour, Mohamed Haouari. 2018. “Robust integrated maintenance aircraft routing and crew pairing.” *Journal of Air Transport Management* 73: 15–31

Bureau of Transportation Statistics. 2019. “Understanding the Reporting of Causes of Flight Delays and Cancellations.” Last modified March 5, 2020.

Burke, Edmund K., Patrick De Causmaecker, Geert De Maerea, Jeroen Mulder, Marc Paelinck, and Greet Vanden Berghe. 2010. “A multi-objective approach for robust airline scheduling.” *Computers & Operations Research* 37 (5): 822–832.

Cacchiani, Valentina, and Juan José Salazar-González. 2017. “Optimal solutions to a real-world integrated airline scheduling problem.” *Transportation Science* 51 (1): 250–268.

Cacchiani, Valentina, and Juan José Salazar-González. 2020. “Heuristic approaches for flight retiming in an integrated airline scheduling problem of a regional carrier.” *Omega* 91.

Cook, Gerald N., and Jeremy Goodwin. 2008. “Airline Networks: A Comparison of Hub-and-Spoke and Point-to-Point Systems.” *Journal of Aviation/Aerospace Education & Research* 17 (2): 51–60.

Cordeau, Jean-François, Goran Stojković, François Soumis, and Jacques Desrosiers. 2001. “Benders decomposition for simultaneous aircraft routing and crew scheduling.” *Transportation Science* 35 (4): 375–388.

Dück, Viktor, Lucian Ionescu, Natalia Kliewer, and Leena Suhl. 2012. “Increasing stability of crew and aircraft schedules.” *Transportation Research Part C: Emerging Technologies* 20 (1): 47–61.

Dunbar, Michelle, Gary Froyland, and Cheng-Lung Wu. 2012. “Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework.” *Transportation Science* 46 (2): 204–216.

Dunbar, Michelle, Gary Froyland, and Cheng-Lung Wu. 2014. “An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing.” *Computers & Operations Research* 45: 68–86.

Eltoukhy, Abdelrahman , Felix Chan, Sai-Ho Chung. 2017. “Airline schedule planning: A review and future directions.” *Industrial Management & Data Systems* 117 (6): 1201–1243.

Fourer, Robert, David M. Gay, and Brian W. Kernighan. 2002. AMPL: a modeling language for mathematical programming. 2nd ed. USA: Duxbury Press/Brooks/Cole Publishing Company.

Gu, Zonghao, Ellis L.Johnson, George L.Nemhauser, and YinhuaWang. 1994. “Some properties of the fleet assignment problem”. *Operations Research Letters* 15 (2): 59-71.

Haouari, Mohamed, Farah Zeghal Mansour, Hanif D. Sherali. 2019. “A New Compact Formulation for the Daily Crew Pairing Problem.” *Transportation Science* 53 (3): 811-828.

IBM. “IBM CPLEX Optimizer.” Accessed May 20, 2020. https://www.ibm.com/analytics/cplex-optimizer

Ionescu, Lucian, and Natalia Kliewer. 2011. “Increasing flexibility of airline crew schedules.” *Procedia-Social and Behavioral Sciences* 20: 1019–1028.

Mercier, Anne, Jean-François Cordeau, and François Soumis. 2005. “A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem.” *Computers & Operations Research* 32 (6): 1451–1476.

Papadakos, Nikolaos. 2009. “Integrated airline scheduling.” *Computers & Operations Research* 36 (1): 176-195.

Parmentier, Axel. 2013. “Aircraft routing: complexity and algorithms.” Rapport de stage de master, École des Ponts ParisTech.

Rosenberger, Jay M., Ellis L. Johnson, and George L. Nemhauser. 2004. “A robust fleet assignment model with hub isolation and short cycles.” *Transportation Science* 38 (3): 357–368.

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

Sandhu, Rivi, and Diego Klabjan. 2007. “Integrated airline fleeting and crew-pairing decisions.” *Operations Research* 55 (3): 439-456.

Shao, Shengzhi, Hanif D. Sherali, and Mohamed Haouari. 2015. “A Novel Model and Decomposition Approach for the Integrated Airline Fleet Assignment, Aircraft Routing, and Crew Pairing Problem.” *Transportation Science* 51 (1): 233-249.

Sherali, Hanif D., Ki-Hwan Bae, and Mohamed Haouari. 2010. “Integrated airline schedule design and fleet assignment: Polyhedral analysis and benders' decomposition approach.” *INFORMS Journal on Computing* 22 (4): 500-513.

Sherali, Hanif D and Warren P. Adams. 1990. “A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems.” *SIAM Journal on Discrete Mathematics* 3 (3): 411-430.

Sherali, Hanif D and Warren P. Adams. 1994. “A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems.” *Discrete Applied Mathematics* 52 (1): 83-106.

Smith, Barry C., and Ellis L. Johnson. 2006. “Robust airline fleet assignment: Imposing station purity using station decomposition.” *Transportation Science* 40 (4): 497–516.

Tekiner, Hatice, S. Ilker Birbil, and Kerem Bülbül. 2009. “Robust crew pairing for managing extra flights.” *Computers & Operations Research* 36 (6): 2031–2048.

Victor, Daniel, and Matt Stevens. 2017. “United Airlines Passenger Is Dragged From an Overbooked Flight.” *New York Times*, April 10, 2017.

Zeghal Mansour, Farah, Mohamed Haouari, Hanif D. Sherali, and Najla Aissaoui. 2011. “Flexible aircraft fleeting and routing at TunisAir.” *Journal of the Operational Research Society*, 62(2): 368-380.

Appendix A: Linear model

Sets, parameters and decision variables	
Sets:	
L	set of daily flights
F	set of aircraft families
K^f	set of aircraft types from family f
S^k	set of maintenance stations for aircraft of type k
A	set of arcs for aircraft routing graph
V	set of nodes
δ_j^-	set of arcs incident to node $j \in V$ in aircraft routing graph
δ_j^+	set of arcs outgoing from node $j \in V$ in aircraft routing graph
Π	set of itineraries
A_M	set of maintenance arcs
A_{NM}	set of non-maintenance arcs
L^{WAF}	set of wraparound flights
H	set of fare classes
L_D	set of flights that depart from the base station
L_A	set of flights that arrive to the base station
$\{0\}$	dummy node that represents both the start and the end of a pairing
$\bar{\delta}_j^-$	set of arcs incident to node $j \in V$ in crew pairing graph
$\bar{\delta}_j^+$	set of arcs outgoing from node $j \in V$ in crew pairing graph
B^S	set of short crew connections
B^C	set of critical crew connections
Parameters:	
N_k	number of aircraft of type k
tj	flying time of flight j
T_j^D	departure time of flight j
T_j^A	arrival time of flight j
S_j^D	departure station of flight j
S_j^A	arrival station of flight j
T_T	turnaround time

T_M	time needed to perform the maintenance check
C_{hk}	passenger seat capacity for fare class $h \in H$ on aircraft of type $k \in K$
r_{ph}	estimated revenue from one ticket for fare class $h \in H$ on flight $j \in L$ within itinerary $p \in \Pi^j$
$\bar{\pi}_{ph}$	mean demand for fare class $h \in H$ on flight $j \in L$ within itinerary $p \in \Pi^j$
$\bar{c}_{jk}$	fixed cost of assigning an aircraft of fleet type k to flight leg j
o_{jh}	opportunity cost per spilled passenger on flight leg j
T_{max}^{DF}	maximum duty flight duration
$T_{min}^{ST}/T_{max}^{ST}$	minimum/maximum crew sit-time between two consecutive flights within the same duty
T_{max}^{DD}	maximum duty duration, assuming that the longest flight duration is shorter than the maximum duty duration
$T_{min}^{LO}/T_{max}^{LO}$	minimum/maximum layover duration between two consecutive duties within the same pairing
$T_{min}^{DP}/T_{max}^{DP}$	minimum/maximum pairing duration
N_{max}^{L}	maximum number of landings within a duty
N_{max}^{D}	maximum number of duties within a pairing
T_{min}^{AST}	default minimum sit-time for the aircraft
I^a	aircraft connection cushion time
I^c	crew connection cushion time
q_{ij}^a	quadratic penalty for the aircraft connections (i,j) with short buffer time
q_{ij}^c	quadratic penalty for the crew connections (i,j) with short buffer time
R	reward for robust connection both covered by crew pairing and aircraft
T_{max}^{M}	maximum number of flying hours that aircraft can perform without maintenance check

Decision variables:

y_{ij}	binary variable that takes 1 if arc $(i,j) \in B$ is selected, and 0 otherwise
x_{ij}	binary variable that takes 1 if arc $(i,j) \in A$ is selected, and 0 otherwise
u_{jk}	total accumulated flying hours for aircraft of type $f \in F, k \in K^f$ since its last maintenance check after serving flight leg $j \in L$
w_{jk}	binary variable that equals 1 if flight leg $j \in L$ is assigned to an aircraft of type $f \in F, k \in K^f$, and 0 otherwise
N_{ph}^{PAS}	number of passengers flying within fare class $h \in H$ and itinerary $p \in \Pi$

z_{ij}	binary variable that takes 1 if a crew follows an aircraft on a connection $(i,j) \in A \cap B$
N_{jf}^{L}	total number of landing for a crew that is eligible to an aircraft family $f \in F$ after serving flight leg $j \in L$
T_{jf}^{DF}	total accumulated duty flight duration for a crew that is eligible for an aircraft family $f \in F$ after serving flight leg $j \in L$
T_{jf}^{DD}	total accumulated duty duration for a crew that is eligible to an aircraft family $f \in F$ after serving flight leg $j \in L$
N_{jf}^{D}	total accumulated number of duties for a crew that is eligible for an aircraft family $f \in F$ after serving flight leg $j \in L$
T_{jf}^{DP}	total accumulated duration of pairing for a crew that is eligible to an aircraft family $f \in F$ after serving flight leg $j \in L$
d_{jf}	integer variable that corresponds to the duration (in days) of the crew pairing that is eligible to an aircraft family $f \in F$, that ends with flight j, $j \in L$
α_{ij}	artificial variable for aircraft routing $(i,j) \in A$
β_{ij}	artificial variable for number of landings within a duty $(i,j) \in B$
γ_{ij}	artificial variable for flying time within a duty $(i,j) \in B$
μ_{ij}	artificial variable for duty duration $(i,j) \in B$
φ_{ij}	artificial variable for number of duties within a pairing $(i,j) \in B$
ω_{ij}	artificial variable for pairing duration $(i,j) \in B$

Linear model:

$$Maximize \sum_{j \in L} \sum_{p \in \Pi_j} \sum_{h \in H} r_{ph} N_{ph}^{PAS} - \sum_{j \in L} \sum_{f \in F} \sum_{k \in K^f} c_{jk} w_{jk} + \sum_{(i,j) \in B \setminus B^S} R z_{ij} - \sum_{(i,j) \in A} q_{ij}^{a} x_{ij} - \sum_{(i,j) \in B^C} q_{ij}^{c} \left(y_{ij} - z_{ij} \right) \tag{L.1}$$

$$\sum_{f \in F} \sum_{k \in K^f} w_{jk} = 1, \forall j \in L \tag{L.2}$$

$$\sum_{(i,j) \in \delta_j^- \cap A^k} x_{ij} = w_{jk}, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{L.3}$$

$$\sum_{(j,i)\in\delta_j^+\cap A^k} x_{ji} = w_{jk}, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{L.4}$$

$$\sum_{(j,i)\in A^k} \alpha_{ji}^k = t_j \sum_{(i,j)\in A^k} x_{ij} + \sum_{(i,j)\in A_{NM}^k} \alpha_{ij}^k, \qquad \forall f \in F, k \in K^f, j \in L_k \tag{L.5}$$

$$t_j x_{ji} \le \alpha_{ji}^k \le \left(T_{max,f}^M - t_i\right) x_{ji}, \qquad \forall f \in F, k \in K^f (j,i) \in A_{NM}^k \tag{L.6}$$

$$t_j x_{ji} \le \alpha_{ji}^k \le T_{max,f}^M x_{ji}, \qquad \forall f \in F, k \in K^f, (j,i) \in A_M^k \tag{L.7}$$

$$\sum_{(i,j)\in A_M^2\cup A_M^4} x_{ij} \le N_k - \sum_{j\in L^{WAF}} w_{jk}, \qquad \forall f \in F, k \in K^f \tag{L.8}$$

$$\sum_{h\in H}\sum_{p\in\Pi_j} N_{ph}^{PAS} \le \sum_{f\in F}\sum_{k\in K^f}\sum_{h\in H} \tilde{C}_{jkh} w_{jk}, \qquad \forall j \in L \tag{L.9}$$

$$\tilde{C}_{jkh} \equiv \min\left\{C_{hk}, \sum_{p\in\Pi_j} \bar{\pi}_{ph}\right\}, \ \forall j \in L, h \in H, f \in F, k \in K^f$$

$$0 \le N_{ph}^{PAS} \le \tilde{\pi} \equiv \min\left\{\bar{\pi}_{ph}, \max_{k\in K^f, f\in F} C_{hk}\right\}, \qquad \forall j \in L, p \in \Pi_j, h \in H \tag{L.10}$$

$$\sum_{(i,j)\in\bar{\delta}_j^-\cap B^f} y_{ij} = \sum_{k\in K^f}\sum_{(i,j)\in\delta_j^-\cap A^k} x_{ij}, \qquad \forall j \in L, f \in F \tag{L.11}$$

$$\sum_{(j,i)\in\bar{\delta}_j^+\cap B^f} y_{ji} = \sum_{k\in K^f}\sum_{(j,i)\in\delta_j^+\cap A^k} x_{ji}, \qquad \forall j \in L, f \in F \tag{L.12}$$

$$\sum_{(i,j)\in B_D^f} y_{ij} - \sum_{(j,i)\in B_A^f} y_{ji} = 0, \qquad \forall f \in F \tag{L.13}$$

$$\sum_{(j,i)\in B^f} \beta_{ji}^f = 1 + \sum_{(i,j)\in B_1^f} \beta_{ij}^f, \qquad \forall j \in L, f \in F \tag{L.14}$$

$$y_{ij} \leq \beta_{ij}^{f} \leq \left(N_{max,f}^{L} - 1\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \tag{L.15}$$

$$y_{ij} \leq \beta_{ij}^{f} \leq N_{max,f}^{L} y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \cup B_A^f \tag{L.16}$$

$$\sum_{f \in F} \sum_{(j,i) \in B^f} \gamma_{ji}^{f} = t_j + \sum_{f \in F} \sum_{(i,j) \in B_1^f} \gamma_{ij}\,, \qquad \forall j \in L \tag{L.17}$$

$$t_i y_{ij} \leq \gamma_{ij}^{f} \leq \left(T_{max,f}^{DF} - t_j\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \tag{L.18}$$

$$t_i y_{ij} \leq \gamma_{ij}^{f} \leq T_{max}^{DF} y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \cup B_A^f \tag{L.19}$$

$$\sum_{f \in F} \sum_{(j,i) \in B^f} \mu_{ji}^{f} = \sum_{f \in F} \sum_{(i,j) \in B_1^f} T_{ijf}^{ST}\, y_{ij} + t_j + \sum_{f \in F} \sum_{(i,j) \in B_1^f} \mu_{ij}^{f}\,, \qquad \forall j \in L \tag{L.20}$$

$$t_i y_{ij} \leq \mu_{ij}^{f} \leq \left(T_{max,f}^{DD} - T_{ijf}^{ST} - t_j\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \tag{L.21}$$

$$t_i y_{ij} \leq \mu_{ij}^{f} \leq T_{max,f}^{DD} y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \cup B_A^f \tag{L.22}$$

$$\sum_{f \in F} \sum_{(j,i) \in B^f} \varphi_{ji}^{f} = \sum_{f \in F} \sum_{(i,j) \in B_D^f \cup B_2^f} y_{ij} + \sum_{f \in F} \sum_{(i,j) \in B_1^f \cup B_2^f} \varphi_{ij}^{f}\,, \qquad \forall j \in L \tag{L.23}$$

$$y_{ij} \leq \varphi_{ij}^{f} \leq (N_{max,f}^{D} - 1) y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \tag{L.24}$$

$$y_{ij} \leq \varphi_{ij}^{f} \leq N_{max,f}^{D} y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \cup B_A^f \tag{L.25}$$

$$\sum_{f \in F} \sum_{(j,i) \in B^f} \omega_{ji}^f = \sum_{f \in F} \sum_{(i,j) \in B_1^f} T_{ij,f}^{ST}\, y_{ij} + \sum_{f \in F} \sum_{(i,j) \in B_2} T_{ij,f}^{LO} y_{ij} + t_j + \sum_{f \in F} \sum_{(i,j) \in B_1^f \cup B_2^f} \omega_{ij}^f, \qquad \forall j \in L \tag{L.26}$$

$$t_i y_{ij} \le \omega_{ij}^f \le \left(T_{max,f}^{DP} - T_{ijf}^{ST} - t_j\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_1^f \tag{L.27}$$

$$t_i y_{ij} \le \omega_{ij}^f \le \left(T_{max,f}^{DP} - T_{ij,f}^{LO} - t_j\right) y_{ij}, \qquad \forall f \in F, (i,j) \in B_2^f \tag{L.28}$$

$$\omega_{j0}^f + T_{min,f}^{LO} y_{ij} \le 1440 d_{j,f}, \qquad \forall j \in L, f \in F, (j,0) \in B_A^f \tag{L.29}$$

$$\sum_{(j,0) \in B_A^f} d_{jf} \le N_f^{crew}, \qquad \forall f \in F \tag{L.30}$$

$$T_{min}^{DP} y_{j0} \le \omega_{j0}^f \le T_{max,f}^{DP} y_{j0}, \qquad \forall f \in F, (j,0) \in \bar{\delta}_j^+ \cap B_A^f \tag{L.31}$$

$$y_{ij} \le x_{ij}, \qquad \forall (i,j) \in B^S \cap A \tag{L.32}$$

$$0 \le z_{ij} \le x_{ij}, \qquad \forall (i,j) \in B \cap A \tag{L.33}$$

$$0 \le z_{ij} \le y_{ij}, \qquad \forall (i,j) \in B \cap A \tag{L.34}$$

$$(x, y, z, w) \text{ binary} \ge 0 \tag{L.35}$$

$$(\alpha, \beta, \gamma, \mu, \varphi, \omega, u, d, N^{PAS}, N^L, T^{DF}, T^{DD}, N^D, T^{DP}) \text{ integer} \ge 0 \tag{L.36}$$

Appendix B: AMPL model

File .mod:

```
/* ###########             Sets                 ############# */

set L;                      #set of daily flights
set F;                      #set of aircraft families
set K {F};             #set of aircraft types
set S {f in F};             #set of maintenance stations
set PI {L};                 #set of itineraries, with subsets that
include flight L
set H;                      #set of all fair classes
set origin = {0};           #dummy node 0
set Base_st;

/* ###########    Parameters                  ############ */

param S_D {j in L} symbolic ;              #departure station of
flight j
param S_A {j in L} symbolic ;              #arrival station of
flight j
set L_D =setof {j in L:S_D[j] in Base_st}(j) ;         #the set of
flights that depart from the base station
set L_A =setof {j in L: S_A[j] in Base_st}(j) ;        #the set of
flights that arrive to the base station
param N {f in F, k in K[f]} >= 0;  #number of aircraft of type k
param t {j in L} >= 0, < 1440;     #flying time of each flight j
param T_D {j in L} >= 0, <= 1440;  #departure time of each flight
param T_A {j in L} >= 0, <= 1440;  #arrival time of each flight
param T_M {f in F, k in K[f]} >= 0;        #time needed to
perform the maintenance check
param T_M_max {f in F} >= 0;               #maximum number of
flying hours without the maintenance check
param T_T {f in F, k in K[f]} >= 0;        #turnaround time
param T_ST_min {f in F} >= 0;              #minimum sit-time
param T_ST_max {f in F} >= 0;              #maximum sit-time
param T_ST {f in F} >= 0;       #default sit-time
param T_DF_max >= 0;            #maximum duty flight duration
param T_DD_max >= 0;            #maximum duty duration
param T_LO_min {f in F} >= 0;   #minimum layover duration
param T_LO_max >= 0;            #max layover duration
param T_DP_min >= 0;            #min duration of pairing
param T_DP_max {f in F} >= 0;   #max duration of pairing
param N_L_max {f in F} >= 0;#max number of landings within a duty
param N_D_max {f in F} >= 0;#max number of duties within a pairing
param tau default 1.2;          # fraction of duty duration
param I_a_cush >= 0;            #aircraft connection cushion time
param I_c_cush >= 0;            #crew connection cushion time
param R >= 0;    #reward if crew follows aircraft on connection
param minutes := 1440;
param factor_crew default 1.5;
```

```
set L_WAF =setof{j in L: T_D[j]>T_A[j]}(j);       #set of
wraparound flights

/*  ################### Aircraft graph   ######################*/

set A1 = setof {i in L, j in L, f in F, k in K[f]:
      S_A[i] == S_D[j] &&
      S_A[i] in S[f] &&
      T_A[i] + T_M[f,k] <= T_D[j]} (i,j,k);

set A31 = setof {i in L, j in L, f in F, k in K[f]:
      (S_A[i] == S_D[j]) &&
       (S_A[i] not in S[f])&&
      (T_A[i] + T_T[f,k] <= T_D[j])} (i,j,k);

set A32 = setof {i in L, j in L, f in F, k in K[f]:
      (S_A[i] == S_D[j]) &&
       (T_D[j] < T_A[i] + T_M[f,k])  &&
      (T_A[i] + T_T[f,k] <= T_D[j])} (i,j,k);

set A3= A31 union A32;
set A41 = setof {i in L, j in L, f in F, k in K[f]:
      S_A[i] == S_D[j] &&
       (T_D[j] + 1440 < T_A[i] + T_M[f,k])  &&
      T_D[j] < T_A[i] + T_T[f,k] <= T_D[j] + 1440} (i,j,k);

set A42=setof {i in L, j in L, f in F, k in K[f]:
      S_A[i] == S_D[j] &&
      (S_A[i] not in S[f])   &&
      T_D[j] < T_A[i] + T_T[f,k] <= T_D[j] + 1440} (i,j,k);

set A4= A41 union A42;
set A2 = setof {i in L, j in L, f in F, k in K[f]:
      (S_A[i] == S_D[j]) &&
      (S_A[i] in S[f]) &&
      (T_D[j] < T_A[i] + T_M[f,k] <= T_D[j] + 1440)&& (i,j,k) not
in A3} (i,j,k);

set A = A1 union A2 union A3 union A4;#set of arcs  A1 union A2
union
set A_M = A1 union A2;
set A_NM = A3 union A4;

/* ###################  Crew graph #####################*/
set B_1prime = setof { i in L, j in L, f in F:
      S_A[i] == S_D[j] && (t[i] + t[j] <= T_DF_max) } (i,j,f);
## initial definition of sit crew arcs, before refining

set B_2prime = setof { i in L, j in L, f in F:
      S_A[i] == S_D[j] && (S_A[i] not in Base_st)} (i,j,f);
## initial defintion of crew layover arcs

param sit_time {(i,j,f) in B_1prime} :=
      if (T_A[i] - T_ST_min[f] <= T_D[j])
      then (T_D[j] - T_A[i])
      else (T_D[j] - T_A[i] + 1440); ## actual sit time
```

```
param layover_time {(i,j,f) in B_2prime} :=
      if (T_A[i] + T_LO_min[f] <= T_D[j] <= T_A[i] + T_LO_max)
      then (T_D[j] - T_A[i])
      else if  (T_A[i] + T_LO_min[f] <= T_D[j] + 1440 <= T_A[i] +
T_LO_max)
      then (T_D[j] - T_A[i] + 1440); ## actual layover time

set B1 = setof {(i,j,f) in B_1prime:
      (t[i] + t[j] + sit_time[i,j,f] <= T_DD_max) &&
      (T_ST_min[f] <= sit_time[i,j,f] <= T_ST_max[f])} (i,j,f);
## set of sit in arcs

set B2 = setof {(i,j,f) in B_2prime:
      (T_LO_min[f] <= layover_time[i,j,f] <= T_LO_max) &&
      (t[i] + t[j] + layover_time[i,j,f] <= T_DP_max[f]) &&
      (i,j,f) not in B1} (i,j,f);
## set of layover arcs

set BD = setof {i in origin, j in L_D, f in F} (i,j,f);
set BA = setof {i in L_A, j in origin, f in F} (i,j,f);
set BS  = setof {(i,j,f) in B1:
      T_ST_min[f] <= sit_time[i,j,f] < T_ST[f]} (i,j,f);

set B = B1 union B2 union BD union BA;#set of crew pairing arcs
set BC within B1 =         #set of critical connections within B1
&& not within BS
      setof {(i,j,f) in B1:
      T_ST[f] <= sit_time[i,j,f] < I_c_cush &&
      (i,j,f) not in BS} (i,j,f);

/* ###############   Cost definition  ################# */

param C {h in H, f in F, k in K[f]} >= 0;   #seat capacity of
aircraft of type k
param pi_mean {j in L, p in PI[j], h in H} default 0 >= 0;  #mean
demand for fare class h within itinerary p
param r {j in L, p in PI[j], h in H} default 0 >= 0;
      #estimated revenu for one ticket for fare class h within
itinerary p
param c_mean {j in L, f in F, k in K[f]} >= 0;   #fixed cost of
assigning an aircraft k to flight j

param o {j in L, h in H} =                  #opportunity cost per
spilled passenger on flight leg j
      if (sum {p in PI[j]} pi_mean[j,p,h] = 0)
      then (0)
      else (0.2 * (sum {p in PI[j]} (r[j,p,h] * pi_mean[j,p,h]))
      / sum {p in PI[j]} (pi_mean[j,p,h]));

param c {j in L, f in F, k in K[f]} :=      #fleet assignment cost
      c_mean[j,f,k] +
      sum {h in H} (o[j,h] *
      max(0,(sum {p in PI[j]} pi_mean[j,p,h] - C[h,f,k])));

param C_wave {j in L, f in F, k in K[f], h in H} :=
```

```
     min(C[h,f,k], (sum{p in PI[j]} pi_mean[j,p,h]));

/*  ########## Penalties & rewards definitions ########### */

param I_a {f in F, k in K[f], (i,j,k) in A} :=   #aircraft planned
idle time
     if (T_D[j] >= T_A[i] + T_T[f,k])
     then (T_D[j] - T_A[i] - T_T[f,k])
     else (T_D[i] + 1440 - T_A[i] - T_T[f,k]);

param q_a {f in F, k in K[f], (i,j,k) in A} := #aircraft quadratic
penalty
     if (I_a[f,k,i,j] < I_a_cush)
     then ((I_a_cush - I_a[f,k,i,j])^2)
     else 0;

param I_c {f in F, (i,j,f) in B1} :=  #crew planned idle time
     if (T_D[j] >= T_A[i] + T_ST_min[f])
     then (T_D[j] - T_A[i] - T_ST_min[f])
     else (T_D[j] + 1440 - T_A[i] - T_ST_min[f]);


param q_c {f in F, (i,j,f) in BC} :=  #crew quadratic penalty
     (I_c_cush - I_c[f,i,j])^2;

/* #############    Variables  ###############*/

var x {(i,j,k) in A} binary;               #1 if arc a is
selected
var y {(i,j,f) in B} binary;               #1 if arc b is
selected
var z {(i,j,f) in B} binary;         #1 if crew follows aircraft
on connection c
var w {j in L, f in F, k in K[f]} binary;        #1 if flight leg
j assigned to aircraft type k
var u {j in L, f in F, k in K[f]} integer >= 0;        #acccum
flying hours since last mntnce check
var N_PAS {j in L, p in PI[j], h in H} integer >=0;    #number of
passengers flying within fare class h and itinerary p

var alpha {A}  integer>= 0;#artificial variable
var beta {B} integer>= 0;  #artificial variable
var gamma {B} integer >= 0;#artificial variable
var mu {B}  integer>= 0;   #artificial variable
var phi {B} integer >= 0;  #artificial variable
var omega {B}  integer>= 0;#artificial variable
var d {i in L_A, f in F, (i,j,f) in BA} >= 0;         #duration
(in days) of the crew pairing

/* ########### Model starts here ################# */

maximize Total_Profit:
     sum {j in L, p in PI[j], h in H} (r[j,p,h] * N_PAS[j,p,h]) -
     sum {j in L, f in F, k in K[f]} (c[j,f,k] * w[j,f,k]) +
     sum {f in F, k in K[f], (i,j,f) in B diff BS: (i,j,k) in A}
(R * z[i,j,f]) -
```

```
     sum {f in F, k in K[f], (i,j,k) in A} (q_a[f,k,i,j] *
x[i,j,k]) -
     sum {f in F, k in K[f], (i,j,f) in BC: (i,j,k) in A}
(q_c[f,i,j] * (y[i,j,f] - z[i,j,f]));

subject to L2 {j in L}:     #only one aircraft used for each flight
     sum{f in F, k in K[f]}  w[j,f,k] = 1;

subject to L3 {j in L, f in F, k in K[f]}:
     sum{(i,j,k) in A} x[i,j,k] = w[j,f,k];

subject to L4 {j in L, f in F, k in K[f]}:
     sum{(j,i,k) in A} x[j,i,k] = w[j,f,k];

subject to L5 {j in L, f in F, k in K[f]}:
     sum{(j,i,k) in A} alpha[j,i,k] = t[j] * sum{(i,j,k) in A}
x[i,j,k] + sum{(i,j,k) in A_NM} alpha[i,j,k];

subject to L6_1 {f in F, k in K[f], (j,i,k) in A}:
     t[j] * x[j,i,k] <= alpha[j,i,k];

subject to L6_2 {f in F, k in K[f], (j,i,k) in A_NM}:
     alpha[j,i,k] <= (T_M_max[f] - t[i]) * x[j,i,k];

subject to L6_3 {f in F, k in K[f], (j,i,k) in A_M}:
     alpha[j,i,k] <= T_M_max[f] * x[j,i,k];

subject to L7_1 {j in L, f in F, k in K[f], (j,i,k) in A_M}:
     t[j] * x[j,i,k] <= alpha[j,i,k];

subject to L7_2 {j in L, f in F, k in K[f], (j,i,k) in A_M}:
     alpha[j,i,k] <= T_M_max[f] * x[j,i,k];

subject to L8 {f in F, k in K[f]}:
     sum{(i,j,k) in A2 union A4} x[i,j,k] <= N[f,k] - sum{j in
L_WAF} w[j,f,k];

subject to L9 {j in L}:
     sum{h in H, p in PI[j]} N_PAS[j,p,h] <=
     sum{f in F, k in K[f], h in H} (C_wave [j,f,k,h] *
w[j,f,k]);

subject to L10 {j in L, p in PI[j], h in H}:
     0 <= N_PAS[j,p,h] <= min(pi_mean[j,p,h], (max{f in F, k in
K[f]} C[h,f,k]));

/*######### Crew pairing starts here  ##################*/

subject to L11 {f in F,j in L}:
     sum{ (i,j,f) in B } y[i,j,f] = sum{(i,j,k) in A: k in K[f]}
x[i,j,k];

subject to L12 {f in F,j in L }:
     sum{(j,i,f) in B } y[j,i,f] = sum{(j,i,k) in A: k in K[f]}
x[j,i,k];
```

```
subject to L13 {f in F}:
      sum{(i,j,f) in BD} y[i,j,f] - sum{(i,j,f) in BA} y[i,j,f] =
0;

/* Maximum number of landings within a duty*/

subject to L14 {j in L}:
      sum{f in F,(j,i,f) in B} beta[j,i,f] =
      1 + sum{f in F,(i,j,f) in B1} beta[i,j,f];

subject to L15_1 {f in F, (i,j,f) in B1 }:
      y[i,j,f] <= beta[i,j,f];

subject to L15_2 {f in F, (i,j,f) in B1  }:
      beta[i,j,f] <= (N_L_max[f] - 1) * y[i,j,f];

subject to L16_1 {f in F, (i,j,f) in B2 union BA}:
      y[i,j,f] <= beta[i,j,f];

subject to L16_2 {f in F, (i,j,f) in B2 union BA}:
      beta[i,j,f] <= (N_L_max[f]) * y[i,j,f];

/* Maximum flying time within a duty*/

subject to L17 {j in L}:
      sum{f in F,(j,i,f) in B} gamma[j,i,f] =
      t[j] + sum{f in F,(i,j,f) in B1} gamma[i,j,f];

subject to L18_1 {f in F, (i,j,f) in B1}:
      t[i] * y[i,j,f] <= gamma[i,j,f];

subject to L18_2 {f in F, (i,j,f) in B1}:
      gamma[i,j,f] <=  (T_DF_max - t[j]) * y[i,j,f];

subject to L19_1 {f in F, (i,j,f) in B2 union BA}:
      t[i] * y[i,j,f] <= gamma[i,j,f];

subject to L19_2 {f in F, (i,j,f) in B2 union BA}:
      gamma[i,j,f] <= T_DF_max * y[i,j,f];

/* Maximum duty duration */

subject to L20 {j in L}:
      sum{f in F,(j,i,f) in B} mu[j,i,f] =
      sum {f in F,(i,j,f) in B1} (sit_time[i,j,f] * y[i,j,f]) +
      t[j] + sum{f in F,(i,j,f) in B1} mu[i,j,f];

subject to L21_1 {f in F, (i,j,f) in B1}:
      t[i] * y[i,j,f] <= mu[i,j,f];

subject to L21_2 {f in F, (i,j,f) in B1}:
      mu[i,j,f] <= (T_DD_max - sit_time[i,j,f] - t[j]) * y[i,j,f];

subject to L22_1 {f in F, (i,j,f) in B2 union BA}:
      t[i] * y[i,j,f] <= mu[i,j,f];
```

```
subject to L22_2 {f in F, (i,j,f) in B2 union BA}:
      mu[i,j,f] <= T_DD_max * y[i,j,f];

/* Maximum number of duties within a pairing*/

subject to L23 {j in L}:
      sum{f in F,(j,i,f) in B} phi[j,i,f] =
      sum{f in F,(i,j,f) in BD union B2} y[i,j,f] +
      sum{f in F,(i,j,f) in B1 union B2} phi[i,j,f];

subject to L24_1 {f in F, (i,j,f) in B2}:
      y[i,j,f] <= phi[i,j,f];

subject to L24_2 {f in F, (i,j,f) in B2}:
      phi[i,j,f] <= (N_D_max[f] - 1) * y[i,j,f];

subject to L25_1 {f in F, (i,j,f) in B1 union BA}:
      y[i,j,f] <= phi[i,j,f];

subject to L25_2 {f in F, (i,j,f) in B1 union BA}:
      phi[i,j,f] <= N_D_max[f] * y[i,j,f];

/* Maximum pairing duration */

subject to L26 {j in L}:
      sum{f in F,(j,i,f) in B} omega[j,i,f] =
      sum{f in F,(i,j,f) in B1 } (sit_time[i,j,f] * y[i,j,f]) +
      sum{f in F,(i,j,f) in B2} (layover_time[i,j,f] * y[i,j,f]) +
      t[j] + sum{f in F,(i,j,f) in B1 union B2} omega[i,j,f];

subject to L27_1 {f in F, (i,j,f) in B1}:
      t[i] * y[i,j,f] <= omega[i,j,f];

subject to L27_2 {f in F, (i,j,f) in B1}:
      omega[i,j,f] <=  (T_DP_max[f] - sit_time[i,j,f] - t[j]) *
y[i,j,f];

subject to L28_1 {f in F, (i,j,f) in B2}:
      t[i] * y[i,j,f] <= omega[i,j,f];

subject to L28_2 {f in F, (i,j,f) in B2}:
      omega[i,j,f] <=  (T_DP_max[f] - layover_time[i,j,f] - t[j]) *
y[i,j,f];

subject to L29 { f in F, (i,j,f) in BA}:
      omega[i,j,f] + T_LO_min[f] <= 1440 * d[i,f,j];

subject to L31_1 {(i,j,f) in BA}:
      T_DP_min * y[i,j,f] <= omega[i,j,f];

subject to L31_2 { (i,j,f) in BA}:
       omega[i,j,f]<=T_DP_max[f] * y[i,j,f];


/* Number of available crew*/
```

```
subject to L30 :
	sum{(i,j,f) in BA} d[i,f,j] <= factor_crew * sum{j in L, f
in F,k in K[f]} w[j,f,k];#number of crews eligible for aircraft f

/* Crew to follow aircraft*/

subject to L32 {j in L, f in F, k in K[f], (i,j,f) in BS: (i,j,k)
in A}:
	y[i,j,f] <= x[i,j,k];

subject to L33_1 {j in L, f in F, k in K[f], (i,j,f) in B: (i,j,k)
in A}:
	0 <= z[i,j,f];

subject to L33_2 {j in L, f in F, k in K[f], (i,j,f) in B: (i,j,k)
in A}:
	z[i,j,f] <= x[i,j,k];

subject to L34_1 {j in L, f in F, k in K[f], (i,j,f) in B: (i,j,k)
in A}:
	0 <= z[i,j,f];

subject to L34_2 {j in L, f in F, k in K[f], (i,j,f) in B: (i,j,k)
in A}:
	z[i,j,f] <= y[i,j,f];
```

File .run:

```
option solver cplex;
model thesis.mod;
data flights.dat;
data itinerary_generated.dat;
data thesis.dat;
data assign_aircr_cost.dat;
data demand.dat;
data fare.dat;

#option cplex_options "writeprob= thesis.lp";
option cplex_options 'timelimit=18000';
option cplex_options 'mipdisplay=2';
solve;

display Total_Profit > thesis.sol;

printf "number of short connections:\n" > thesis.sol;
display sum{i in L, j in L, f in F: (i,j,f) in BS} y[i,j,f]  >
thesis.sol;

printf "number of critical connections:\n" > thesis.sol;
display sum{(i,j,f) in BC} y[i,j,f]  > thesis.sol;

printf "number of aircraft critical connections:\n" > thesis.sol;
display sum{f in F, k in  K[f], (i,j,k) in A: q_a[f,k,i,j] > 0}
x[i,j,k] > thesis.sol;
```

```
printf "number of crews following the aircraft:\n" > thesis.sol;
display    sum{f in F, k in  K[f], (i,j,f) in B: (i,j,k) in A}
z[i,j,f]  > thesis.sol;

printf "FTC(sittime + layover time):\n" > thesis.sol;
display sum{(i,j,f) in B1} sit_time[i,j,f]*y[i,j,f] + sum{(i,j,f)
in B2}  layover_time[i,j,f]*y[i,j,f]  > thesis.sol;

printf "revenue:\n" > thesis.sol;
display    sum {j in L, p in PI[j], h in H} (r[j,p,h] *
N_PAS[j,p,h]) > thesis.sol;

printf "cost of aircraft assignment:\n" > thesis.sol;
display    sum {j in L, f in F, k in K[f]} (c[j,f,k] * w[j,f,k]) >
thesis.sol;

printf "reward from crew following the aircraft:\n" > thesis.sol;
display    sum {f in F, k in K[f], (i,j,f) in B diff BS: (i,j,k)
in A} (R * z[i,j,f]) > thesis.sol;

printf "penalty for short connection:\n" > thesis.sol;
display    sum {f in F, k in K[f], (i,j,k) in A} (q_a[f,k,i,j] *
x[i,j,k]) > thesis.sol;

printf "penalty for critical connection without crew following
aircraft:\n" > thesis.sol;
display    sum {f in F, k in K[f], (i,j,f) in BC: (i,j,k) in A}
(q_c[f,i,j] * (y[i,j,f]-z[i,j,f])) > thesis.sol;

display _total_solve_system_time > thesis.sol;
display _total_solve_user_time > thesis.sol;

display _solve_system_time > thesis.sol;
display _solve_user_time > thesis.sol;

display w > thesis.sol;
display x > thesis.sol;
display y > thesis.sol;
display z > thesis.sol;
```

Appendix C: Python code

File data_generator.py:

```
file_name = "thesis.dat"
file = open(file_name, "w")
data_file_name = "fleet.txt" #heading should be deleted
my_dict = {}
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        my_dict.setdefault(words[0], []).append(words[1])
text_file.close()
file.write("set F :=")
for k in my_dict:
    file.write(" " + k)
file.write(";")
for k, v in my_dict.items():
    file.write("\nset K["+k+"]:= ")
    for item in v:
        file.write(item)
        file.write(" ")
    file.write(";")
file.write("\n\nset Base_st := ORD IAH LAX EWR SFO IAD DEN CLE;\n")
data_file_name = "family.txt"
my_dict2 = {}
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        for x in range(11,len(words)):
            my_dict2.setdefault(words[0], []).append(words[x])
text_file.close()
for k, v in my_dict2.items():
    file.write("\nset S["+k+"]:= ")
    for item in v:
        file.write(item)
        file.write(" ")
    file.write(";")
file.write("\n\nparam N:=  ")
data_file_name = "fleet.txt"
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        file.write(words[0] + " " + words[1] + " " + words[8] + "\n")
file.write(";")
text_file.close()
file.write("\n\nparam T_T:=    ")
data_file_name = "fleet.txt"
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        file.write(words[0] + " " + words[1] + " " + words[6] + "\n")
file.write(";\n\n")
text_file.close()
file.write("set H := Y W C;\n\n")
```

```
file.write("param C := \n")
data_file_name = "fleet.txt"
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        file.write("[*, "+words[0]+", "+words[1]+"] C " + words[10] + " W
" + words[11] + " Y " + words[12] + "\n")
file.write(";\n\n")
text_file.close()
file.write("param T_DF_max := 480;       #maximum duty flight duration\n"
           "param T_DD_max := 720;       #maximum duty duration\n"
           "param T_DP_min := 300;       #min duration of pairing\n"
           "param T_LO_max := 1980;          #max layover duration\n\n"
           "param: T_ST_min T_ST T_ST_max T_LO_min N_L_max T_DP_max
N_D_max := ")
data_file_name = "family.txt"
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        file.write("\n" + words[0] + " " + words[2] + " " + words[1] +
                   " " + words[3] + " " + words[4] + " " +
                   words[7] + " " + words[8] + " " + words[9])
file.write(";\n\nparam: T_M_max := ")
text_file.close()
dict = {}
data_file_name = "fleet.txt"
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        dict[words[0]] = words[2]
for k in dict:
    file.write("\n" + k + " " + dict[k])
file.write(";\n\n")
text_file.close()
file.write("param T_M := \n")
data_file_name = "fleet.txt"
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        file.write(words[0] + " " + words[1] + " " + words[5] + "\n")
file.write(";\n\n")
text_file.close()
file.write("param I_a_cush := 60;       #aircraft connection cushion
time\n"
            "param I_c_cush := 60;       #crew connection cushion time\n"
            "param R:= 15000;\n")
file.close()
```

File assign_aircr_cost.py:

```
#headingss from the data file should be removed
data_file_name = "flight.txt"
data_file_name2 = "fleet.txt"
data_file_name3 = "assign_aircr_cost.dat"
dict = {}
```

```
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        dict[words[0]] = words[5]
text_file.close()
text_file2 = open(data_file_name2, "r")
text_file3 = open(data_file_name3, "w")
text_file3.write("#fixed cost of assigning an aircraft k to flight j \n
param c_mean:=      ")
for line in text_file2:
    if line != "\n":
        words = line.split()
        text_file3.write("\n[*, "+words[0]+", "+words[1]+"] ")
        for k in dict:
            cost = float(dict[k])*float(words[7])/60
            text_file3.write(k+" "+str(cost)+" ")
text_file3.write(";")
text_file2.close()
text_file3.close()
```

File flights.py:

```
#headings from the data file should be removed
data_file_name = "flight.txt"
list_dep = []
list_arr = []
bases = ['ORD', 'IAH', 'LAX', 'EWR', 'SFO', 'IAD', 'DEN', 'CLE']
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        if words[1] in bases:
            list_dep.append(words[0])
        if words[3] in bases:
            list_arr.append(words[0])
text_file.close()
data_file_name = "flights.dat"
text_file = open(data_file_name, "w")
file_name = "flight.txt"
file = open(file_name, "r")
text_file.write("param: L: S_D    T_D    S_A    T_A    t:=\n")
for line in file:
    if line != "\n":
        words = line.split()
        for x in range(0,6):
            text_file.write(words[x] + " ")
        text_file.write("\n")
text_file.write(";")
file.close()
text_file.close()
```

File itineraries_for_flights.py:

```
data_file_name = "itinerary.txt" #heading should be deleted
my_dict = {}
text_file = open(data_file_name, "r")
for line in text_file:
    if line != "\n":
        words = line.split()
        for x in range(5, len(words)):
            my_dict.setdefault(words[x], []).append(words[0])
```

```
text_file.close()
for k, v in my_dict.items():
    new_list = []
    for item in v:
        if item not in new_list:
            new_list.append(item)
    my_dict[k] = new_list
data_file_name = "itinerary_generated.dat"
text_file = open(data_file_name, "w")
for k, v in my_dict.items():
    text_file.write("\nset PI["+k+"]:= ")
    for item in v:
        text_file.write(item)
        text_file.write(" ")
    text_file.write(";")
text_file.close()
data_file_name = "itinerary.txt"
write_file = "demand.dat"
text_file = open(data_file_name, "r")
file = open(write_file, "w")
file.write("#mean demand for fare class h within itinerary p\n")
file.write("param pi_mean := ")
for line in text_file:
    if line != "\n":
        words = line.split()
        for x in range(5, len(words)):
            file.write(words[x]+" "+words[0]+" "+words[1]+"
"+words[3]+"\n")
file.write(";")
text_file.close()
file.close()
data_file_name = "itinerary.txt"
write_file = "fare.dat"
file = open(write_file, "w")
text_file = open(data_file_name, "r")
file.write("#fare of class \nparam r := ")
for line in text_file:
    if line != "\n":
        words = line.split()
        for x in range(5, len(words)):
            file.write(words[x]+" "+words[0]+" "+words[1]+"
"+words[2]+"\n")
file.write(";")
file.close()
text_file.close()
```