DEVELOPMENT OF A DECISION SUPPORT SYSTEM FOR THE OPTIMIZATION OF RESERVE CREW OPEN-TIME PAIRINGS

by

Fatemeh Mortazavi

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
August 2017

Table of Contents

List of Tables

List of Figures

Abstract

Crew costs constitute the largest direct operating cost after fuel costs for airlines. Even a small percentage of savings in crew costs is significant in monetary value. After initial schedules are published, changes in flight schedules, crew and aircraft availability can result in segments of flights without crew members, which is known as open-time flying. Our study, done in collaboration with Jazz Aviation LP, deals with the re-optimization of the open-time flying for reserve cabin crew. A two-phase approach is proposed: the first phase generates all legal potential pairings and the second phase solves a set partitioning model to select the optimal combination of pairings. This pairing problem includes multiple duty types, multi-day rolling time horizon, complex non-linear crew pay structure, multiple bases, and crew deadheading. Our study addresses the research gap between the monthly and the day of operations problems and designs a decision support system to assist schedulers.

List of Abbreviations Used

ACSP	Airline Crew Scheduling Problem
BIP	Binary Integer Programming
CBA	Collective Bargaining Agreement
CSP	Crew Scheduling Problem
IP	Integer Programming
LP	Linear Programming
SCP	Set Covering Problem
SPP	Set Partitioning Problem
TAFB	Time Away From Base
UTC	Coordinated Universal Time

Glossary

block hour The number of hours a crew member spends on productive flying.

cabin crew This category of crews includes flight attendants and pursers.

check-out The time at which a crew member's duty period / pairing ends.

check-in The time at which a crew member's duty period / pairing begins.

credit hour The units of work that a crew member earns for pay purposes (i.e., the number of hours a crew member gets paid).

crew base In airlines with multiple bases, every crew member is associated with a base which is basically where they live, i.e., their domicile or home city.

deadhead Crew members usually work on flights they are assigned to, but a pairing may also contain so-called deadheads, where the crew member is not working, but is only transported as a passenger from one airport to another.

duty period Legs are grouped into what is called a duty period, which can simply be thought of as a working day for crew members.

layover Duty periods are separated by layovers, which are also referred to as overnight connections or rest periods.

leg A single non-stop flight is usually referred to as a leg or segment. Legs are specified by flight number, origin and destination airport codes as well as local departure and arrival times and dates.

open-time flying A leg that does not have a full crew and needs to be covered by reserve crew.

pairing A crew pairing, also referred to as rotation, is made up of individual duty periods, which are separated by periods of rest (also known as layover or overnight).

purser The flight attendant who is in charge on a flight that requires more than one flight attendant.

reserve crew A crew member who is scheduled to be on call for blocks of days in a month. In other words, a crew member whose calendar month is only composed of reserve availability periods and days off.

sit connection A connection within duty, i.e., between any two legs, is called a sit connection, or simply sit.

Acknowledgements

First and foremost, I would like to thank my supervisors, Dr. Alireza Ghasemi and Dr. Claver Diallo, for their valuable academic inputs, support, help and encouragements. I cannot emphasize enough how supportive and caring they have been towards me. They have helped me grow, develop a more positive outlook towards anything and everything, believe in myself and feel confident that I can take up challenges.

I would also like to acknowledge Dr. Majid Taghavi of the School of Health Administration at Dalhousie University and Dr. Peter Vanberkel of the Department of Industrial Engineering at Dalhousie University as readers of this thesis. I am grateful to them for their valuable comments.

I would also like to thank Jazz Aviation LP for funding and inspiring this research project. I feel fortunate to have worked on a collaborative project between the Industrial Engineering Department at Dalhousie University and the airline, from which I would especially like to thank the project leader, Lisa Vad, and the support team, Daryl Joseph and Josh MacKay, for their incredible support, encouragement, patience and time as well as for providing an inclusive environment.

I would like to express my sincere gratitude to my mother and my father whose wisdom, love, support and prayers always bring me comfort. I am grateful for having them by my side. I am also thankful to my brother for being the caring big brother he is, and for sharing his life experiences yet encouraging me to be open to new experiences.

I would also like to show my appreciation to my many good friends, from near and far, who have always been there for me as if we were family. I would especially like to thank my friend, Elna Siebring, for her friendship, having answers to my many questions and being a role model that I know I can look up to on the journey of life.

Chapter 1

Introduction

This thesis presents the develpment of a decision support system for the optimization of reserve crew open-time pairings for Jazz Aviation LP. In this chapter, resource planning and scheduling for airlines is introduced in Section 1.1. The airline crew scheduling problem is explored in Section 1.2 and Section 1.3 describes the motivation for this study.

1.1 Resource Planning and Scheduling for Airlines

The main resources of an airline are aircraft and crews. As a result, the planning and scheduling of these resources requires various departments to collaborate and perform complex tasks including the construction of the flight timetable and fleet assignment with regards to aircraft, followed by the scheduling of cockpit and cabin crews (Gopalakrishnan & Johnson [23]). The creation of the flight timetable is a logical prerequisite for other resource planning problems. Flight timetable construction intends to match market demand with the available fleets, considering external restrictions such as available time slots for every airline at different airports (Andersson, Housos, Kohl & Wedelin [5]). The output of this stage is a number of non-stop flights, called flight segments or legs, that the airline plans to operate (Andersson et al. [5]).

The next stage in resource planning is known as the Fleet Assignment Problem where aircraft, distinguished by fleet type and tail number, are allocated to the flight legs. As with the production of the flight timetable, some airports may impose restrictions and not permit the operation of certain aircraft (Andersson et al. [5]). The main concern in Fleet Assignment is to check whether the constructed timetable can feasibly be operated using the aircraft at hand (Andersson et al. [5]). Since the expected revenue of a flight depends on the number of passenger seats, therefore, following feasibility checks, the objective of this stage is to maximize profit, i.e., expected revenue minus operating costs (Andersson et al. [5]).

Once the airline has decided which flight segments to operate using the available fleet, the Crew Scheduling Problem (CSP) is tackled. The objective of the airline CSP is to find an assignment of flight crews to a given flight schedule that minimizes crew cost (Vance, Barnhart, Johnson & Nemhauser [34]). The airline CSP is an extensive and complex problem that is the focus of this research. The purpose is to design and develop a decision support system to assist crew schedulers of a local airline. In addition, a literature review shows the existence of a gap in research between the monthly problem and the day of operations airline crew scheduling problem, which is addressed in this work.

1.2 Airline Crew Scheduling Problem (ACSP)

The objective of an ACSP is to identify sequences of flight segments and to assign cockpit and cabin crews to these sequences, while trying to minimize crew costs or maximize crew utilization and crew satisfaction. The ACSP is generally solved in two stages, namely crew pairing and crew assignment or rostering. Key terms used in the ACSP along with their definitions can be found in the glossary of this thesis. Section 1.2.1 provides more insight into the first stage of the crew scheduling problem, which is the focus of this thesis. The concept of open-time flying is introduced in Section 1.2.2.

1.2.1 Crew Pairing Construction

The process of constructing crew pairings is based on the concepts of flight legs (also known as segments), duty periods, and crew pairings (sometimes referred to as rotations).

A leg or segment is a single non-stop flight. For the purpose of this study, the flight number, origin and destination airport codes as well as local departure and arrival times and dates fully describe a leg. (Note that the reason why equipment types and aircraft tail numbers are not of concern will be explained in Chapter 3.) Table 1.1 shows an example of a leg with flight number 7066 that has been scheduled to depart from YYC (Calgary, AB, Canada) at 21:00 p.m. local time on September 1^{st} 2016 and to arrive in YWG (Winnipeg, MB, Canada) at 00:23 a.m. local time.

Flight No.	Origin Airport	Destination Airport	Departure Date	Local Departure Time	Local Arrival Time
7066	YYC	YWG	09/01/16	2100	0023

Table 1.1: Example of a flight leg

The local arrival date of each flight leg is required to correctly calculate flight durations and other indicators in the context of multi-day planning periods. This piece of information is usually missing in flight schedules. Given the local departure date, local departure time, local arrival time, departure airport UTC offset, and arrival airport UTC offset of any leg, the corresponding local arrival date can be determined using the pseudo-code given in Appendix A.

Duty periods are sequences of flight segments with only brief periods between them to allow for connecting between flights (known as sit connection or sit). In simple terms, a duty period can be thought of as a working day for flight crew members (Vance et al. [34]).

Pairings are made up of individual duty periods, which are separated by periods of rest (also known as layover or overnight). In other words, a crew pairing or rotation is a sequence of legs for an unspecified crew member, which begins and ends at the same crew base. Figure 1.1 provides a visual representation of how flight segments, duty periods, sit connections and rest periods come together to form a pairing that begins and terminates at the same crew base. Airlines may have one or several crew bases. In airlines with multiple bases, every crew member is associated with a base which is typically their domicile or home city. In order for crews to start and finish pairings at the same base, it may be necessary to use deadheads. Deadheads refer to flying crew members as passengers from their home base to another base/city where they would start their duties. Deadheads also occur when crew members are flown as passengers at the end of a pairing from a base/city to their home base. Apart from association with a particular base, crew members belong to one of two main categories, namely cockpit crew (or pilots) and cabin crew (or flight attendants). In general, cabin crew pairings are independent of cockpit crew pairings.

For duty periods and pairings to be considered legal, they must follow a lengthy list of feasibility rules, resulting from government regulations and collective bargaining agreements (CBA) which differ for cockpit and cabin crew. Thus, approaching the crew pairing problem for flight attendants, for instance, which is the subject of this work, requires the use of work rules for this crew category. Moreover, at some airlines, crews may be further grouped into regular and reserve crew positions for an arbitrary calendar month which requires different scheduling processes. The rules governing the entire crew scheduling problem, including pairing construction, is highly dependent on the airline. As a result, although techniques for generating crew pairings are well established, they are highly customized and require significant implementation efforts to adapt to each airline. Depending on the complexity of the requirements and the number of flights considered, certain CSP modeling methods have proven to be more appropriate than others. It is therefore very important to clearly understand the operating environment of an airline in order to identify the best crew scheduling method and adapt it to solve the problem at hand.

1.2.2 Open-time Pairings

Existing papers have mostly examined the CSP at the strategic level, specifically focusing on modeling and finding near-optimal or optimal solutions for the typical pre-month planning process. On the other end of the planning spectrum, lies the crew rescheduling problem at the operational level, which aims to recover crew schedules from disruptions that may occur during daily operations. Compared to the monthly problem, the crew rescheduling problem is a less-studied area, although after initial

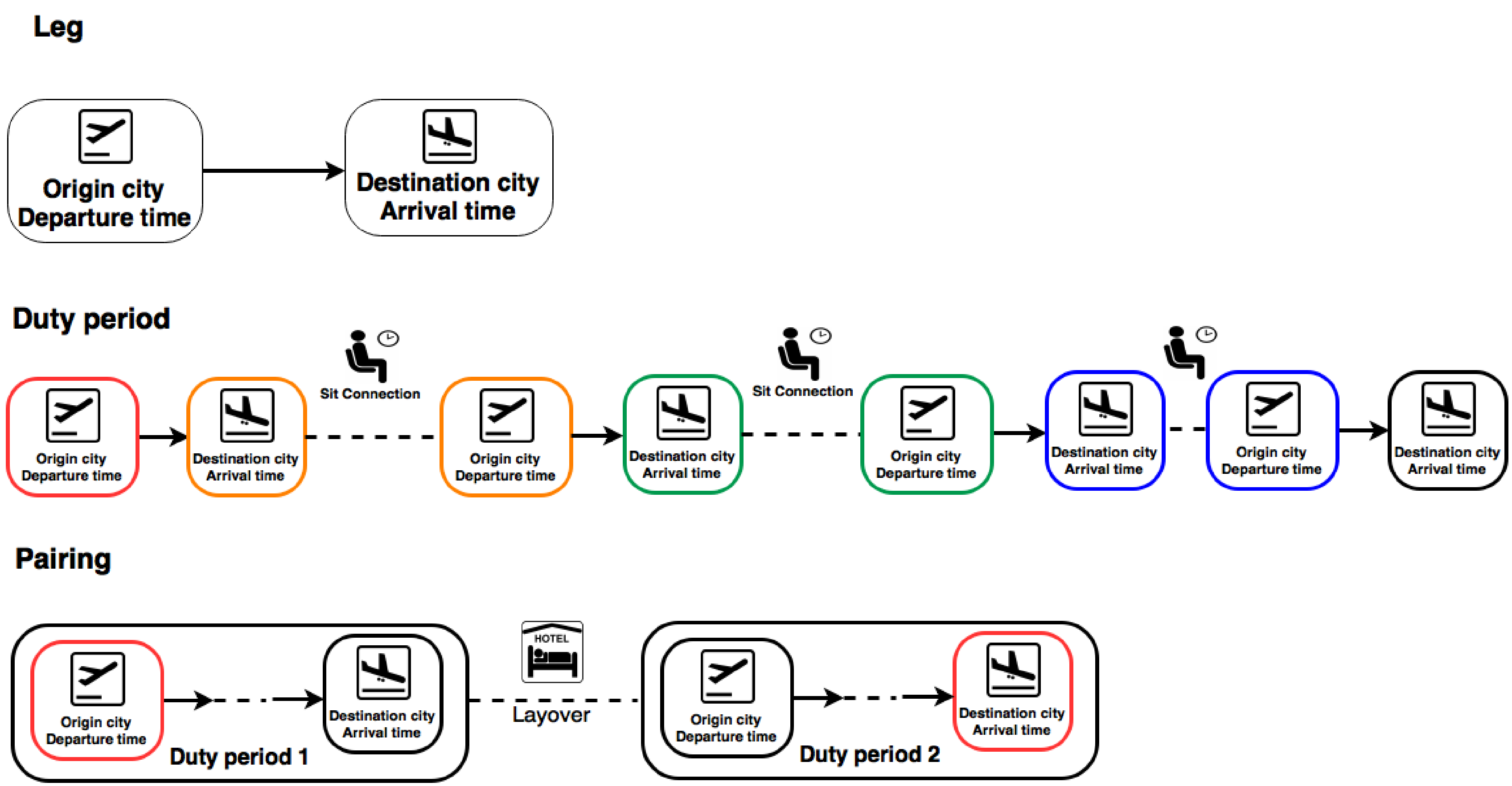


Figure 1.1: Visual representation of flight leg, duty period and pairing

schedules are published, changes in crew availability, flight schedules, and aircraft are inevitable, resulting in what is known as open-time flying, and thus, a proper approach to this time-critical problem can prevent unnecessary extra costs for airlines. Previous research in this area have mainly addressed the rescheduling problem encountered on the day of operations. However, open-time flying may be caused by various sources, some of which make it possible to identify broken pairings several days prior to the day of operations, therefore allowing for a significant portion of disruptions to be dealt with in a less time-restrained planning environment by utilizing reserve crews as efficiently as possible. This aspect of crew rescheduling is overlooked by previous academic research. Our study, done in collaboration with industry partner, Jazz Aviation LP, has led to the design of a two-phase approach for the crew pairing optimization problem with respect to open-time flying for reserve cabin crew.

1.3 Background and Motivation

Jazz Aviation LP (formerly Air Canada Jazz) operates flights on behalf of Air Canada as a contract carrier. This airline operates more flights and flies to more Canadian destinations than any other carrier. They provide service to and from many smaller communities in Canada and the United States under the brand name Air Canada Express, which serves approximately 70 different destinations. They also operate to larger centers at off-peak times as a complement to Air Canada's schedule. According to the Corporate Fact Sheet retrieved on June 22^{nd} 2017 [2], Jazz Aviation has 4,333 employees and operates 722 flights daily, flying a total of 30,000 passengers each day. Their fleet consists of a total of 119 aircraft, including 44 Canadian-made Bombardier Dash 8, 31 CRJ and 44 Q400 NextGen aircraft, which are among the most efficient aircraft flying today [1]. Jazz Aviation's headquarters are located in Halifax, Nova Scotia at the Stanfield International Airport, where all of the crew planning and the crew scheduling are done (Wiggins & Conrad [35]). However, Jazz Aviation also has five regional offices and operation bases across Canada in Halifax, Montreal, Toronto, Calgary, and Vancouver.

Flight crews at Jazz Aviation are scheduled on a monthly basis and their flying is grouped into flying blocks, which are composed of pairings (Wiggins & Conrad [35]). Credit losses are incurred when a crew member is paid but is not working for example because of a long break between two legs. At Jazz Aviation, it is required that the total credit hours of each flying block fall within a specific window (Wiggins & Conrad [35]). According to the collective agreement, flying blocks are assigned through a bidding process. As it is not possible to fit all flights into blocks that meet the above criteria, some flights remain unassigned (Wiggins & Conrad [35]) or are incomplete (i.e., do not form a complete loop). These are said to be in open-time. In addition, there are various other factors which can cause previously scheduled flights to fall into open-time, including weather disruptions, ground delays, aircraft maintenance issues, crew unavailability due to sickness or fatigue (Wiggins & Conrad [35]).

Currently at Jazz Aviation, open flying is assigned semi-manually by the Crew Scheduling Department. This research is aiming at designing a decision support system to help the Scheduling Department re-optimize the generation of legal open-time pairings for reserve cabin crew. The optimization problem has been complicated by the large number of possible deadheads to be included in creating the pairings. In this research, the creation of optimal pairings for cabin crew flights in open-time is studied. The key characteristics of the crew pairing problem under consideration include multiple duty types, multi-day rolling time horizon, complex non-linear crew pay structure in terms of credit hours, multiple crew bases, and use of actual flights available for deadheading crew from and/or to base. In the case of Jazz Aviation, all flights between two Canadian cities as well as flights originating from any US city served by the airline and its partners which land in Canada are potential deadheads and have been considered.

An overview of relevant research in the area of airline crew scheduling is presented in Chapter 2. As airline requirements vary substantially from one company to another, existing modeling techniques for the crew pairing problem could not be adapted directly. Therefore, a two-phase approach to the open-time pairing optimization problem for reserve cabin crew is proposed, which will be described in Chapter 3. Experiments, results and discussions are presented in Chapter 4. Finally, conclusions and future extensions are given in Chapter 5.

Chapter 2

Literature Review

Crew costs constitute the largest direct operating cost of airlines after fuel costs (Anbil, Gelman, Patty & Tanga [4]; Andersson et al. [5]). Therefore, even small percentage of savings in crew costs add up to significant monetary values for airlines (Anbil et al. [4]; Hoffman & Padberg [24]). Desaulniers et al. [18] have mentioned that for major airlines, "...a 1 % decrease in the total crew costs often amounts to tens of millions of dollars per year in additional profits." A large portion of flight crew costs is controllable (Anbil et al. [4]), prompting academia and industry to examine the CSP (Bazargan [11], p. 82). Hoffman and Padberg [24] have stated that minimizing the cost of airline crew schedules also results in crews being happier with the schedules because the obtained solutions tend to allocate more time to paid flying than waiting on the ground.

The objective of the airline crew scheduling problem (ACSP) is to find a minimum-cost assignment of flight crews to flight schedules (Vance et al. [34]). Crew scheduling involves the processes of identifying sequences of flight segments and assigning cockpit and cabin crews to these sequences, while trying to minimize crew costs or maximize crew utilization. The CSP is typically solved in two stages: building crew pairings and crew assignment or rostering (Yan & Tu [36]; Bazargan [11], p. 82; AhmadBeygi, Cohn & Weir [3]). A pairing can be viewed as a single-day/multi-day sequence of flights that starts and finishes at the same crew base and "can feasibly be flown [or covered] by a single crew" (AhmadBeygi et al. [3]). From another viewpoint, pairings are composed of individual duty periods, i.e., legal work days, which are separated by rest periods (AhmadBeygi et al. [3]). Therefore, duties are sequences of flight segments that contain brief connection times between consecutive flights (Vance et al. [34]; AhmadBeygi et al. [3]). A lengthy list of rules that restrict pairings and duty periods are present in crew scheduling and are derived from:

- government regulations, collective agreements, operation and safety requirements (AhmadBeygi et al. [3]);
- "sound economics" (Arabeyre, Fearnley, Steiger & Teather [6]);
- "certain company restrictions imposed by the carrier to assure the smooth transition of crews to flights" (Hoffman & Padberg [24]).

Taking these considerations into account results in a highly complex cost function and constraints, which are even difficult to formulate (Balas [8]).

In the first stage of the CSP, namely crew pairing, legal pairings of flight legs, which satisfy governmental regulations and airline-specific collective agreements, are constructed. Generally, the time horizon associated with pairing construction is several days (Desaulniers et al. [18]). For instance, the crew pairing problem at Air France, involving medium-haul flights, which was addressed by Desaulniers et al. [18], has a periodic weekly horizon. On the other hand, AhmadBeygi et al. [3] have focused on formulating the daily problem, where it is assumed that each flight is repeated every day of the week.

The second stage of the CSP, commonly known as the crew assignment or rostering problem, deals with assigning anonymous minimum-cost pairings to named individuals, while satisfying training, vacation and other requirements. For instance, cabin crew must meet the language proficiency qualification for international flights (Kohl & Karisch [25]). In the crew assignment problem, a category of rules addresses individual assignments or rosters (single crew member rules), while another category focuses on combinations of rosters or crew members (multiple crew member rules) (Fahle et al. [22]). The objective of this phase, also referred to as the crew workload assignment problem or the block assignment problem (Desaulniers et al. [18]), is a combination of cost efficiency and crew satisfaction (Kohl & Karisch [25]). Kohl and Karisch [25] have provided a comprehensive description of crew rostering problems in the airline industry and have shown how complex real-world crew rostering is. For instance, they have argued that the maximization of crew satisfaction is an important part of crew rostering (Kohl & Karisch [25]) as overall crew satisfaction may impact the quality and economic return of an airline's operations (El Moudani & Mora-Camino [21]). However, it is very difficult to quantify crew satisfaction in a fair and logical way that would be accepted by the crew (Kohl & Karisch [25]). Other research in airline crew rostering includes the works of Day and Ryan [16], Dawid, Konig and Strauss [15], El Moudani, Cosenza and Mora-Camino [20], Cappanera and Gallo [12], and Maenhout and Vanhoucke [27].

CSP is classified in terms of personnel category as work rules for flight attendants differ from those for pilots. In practice, most airlines divide the crew pairing problem into two independent problems to be solved separately for cabin and cockpit crews (Desaulniers et al. [18]). Vance et al. [34] have considered the problem of scheduling pilots, while Yan and Tu [36] have focused on scheduling flight attendants. Desaulniers et al. [18] have implemented a crew pairing problem optimizer at Air France, where the pilot pairing problem and the flight attendant pairing problem are solved separately. In addition, crew qualifications regarding aircraft types determine whether or not the pairing problem must be further decomposed into individual problems (Gopalakrishnan & Johnson [23]). In terms of problem size, the flight attendant problem tends to be much larger than the pilot problem (Gopalakrishnan & Johnson [23]), because usually, depending on the airline, flight attendants are qualified to work on multiple, if not all, aircraft types, whereas pilots are trained to fly only one type of aircraft. As a result, the pilot scheduling problem is separable by fleet type (Schaefer, Johnson, Kleywegt & Nemhauser [30]).

Airline crew scheduling is performed in several planning levels (Stojković, Soumis & Desrosiers [33]). Crew planning at the strategic level uses the flight timetable that generally spans about one calendar month (Medard & Sawhney [28]) in order to generate feasible crew pairings without considering individual needs or preferences of crews (Maenhout & Vanhoucke [27]) in the first stage, and to assign cockpit and cabin crews to the pairings in the second stage after the pre-assignment of other activities such as training, office duties, or medical checks (Kohl & Karisch [25]). This process is referred to as the pre-month planning process (Sohoni, Johnson, & Bailey [31]) and usually takes place weeks before the flights are operated. At the monthly planning level, a major challenge is problem size, which can be represented by the number of flight legs to cover in the planning horizon. This is why various modelling approaches and solution methods are found in the literature with regards to solving large-scale airline crew scheduling problems (ACSP), which are known to be difficult combinatorial optimization problems (Anbil et al. [4]; Stojković et al. [33]; Deng & Lin [17]). Thus, much effort has been put on overcoming the issue of solving to optimality large integer programs arising within the ACSP in a reasonable amount of time using available software and hardware technologies. Examples of research relevant to strategic crew planning problems include Baker, Bodin, Finnegan and Ponder [7], Hoffman and Padberg [24], Desaulniers et al. [18], Vance et al. [34], Chu, Gelman and Johnson [13], Andersson et al. [5], Yan and Tu [36], AhmadBeygi et al. [3], Deng and Lin [17].

The operational crew planning of an airline includes the so-called time critical crew recovery problems, also known as crew re-scheduling (Nissen & Haase [29]), which address disruptions during daily operations that may be caused by changes in flight schedules, aircraft and crew availability (Medard & Sawhney [28]). For instance, sickness of crew (crew absences), flight delays and cancellations due to severe weather conditions, peak-hour congestion at airports, crew and passenger delays and strikes (Stojković, Soumis, Desrosiers & Solomon [32]) (schedule disruptions), or unplanned aircraft maintenance requirements (aircraft substitutions) are inevitable events that affect the initial schedule (Stojković et al. [33]). To be specific, a flight delay might make it impossible for a crew member to reach the following flight on their scheduled pairing; a flight cancellation caused by aircraft unavailability results in the scheduled pairing assigned to crew to become operationally infeasible; and a crew calling sick leaves the initially assigned pairing uncovered (Medard & Sawhney [28]). Such issues must be dealt with efficiently through crew re-scheduling. The crew recovery problem focuses on finding a reassignment of aircraft and crews that satisfies safety regulations and has little impact on passengers at minimum recovery operation cost (Lettovsky, Johnson & Nemhauser [26]). The objective function may include several goals, such as minimizing passenger delay, returning to the original plan as quickly as possible as well as minimizing passenger inconvenience (non-quantifiable) (Clausen, A. Larsen,

J. Larsen & Rezanova [14]). Among the characteristics of the crew re-scheduling problem are the significantly smaller planning horizon compared to the pre-month planning problem, as well as time sensitivity, which demands a solution technique that takes as little computation time as possible. Medard and Sawhney [28] have suggested that "in the context of day of operations, solutions must be found within $1-5$ minutes." Their work is an example of research in day-of crew re-planning, in which they have formulated the crew recovery problem as a flight-based crew rostering problem, where pairing construction and pairing assignment are done in a single step using a generate-and-optimize approach.

Open-time flying may be a result of various sources. In addition to daily disruptions that may leave flights uncovered, open-time flying is encountered frequently in airlines that use the preferential bidding system to award monthly work schedules based on crew bids. Thus, on the operational level, crew scheduling attempts to legally and optimally match available crews with all such open-time flying. How airlines deal with uncovered flights and broken pairings varies from one airline to another. For instance, some airlines choose to maintain reserve crews in addition to regular crews to provide smooth daily operations (Sohoni, Johnson & Bailey [31]). This is the case at Jazz Aviation LP, the industry partner in this study, where the calendar month for reserve crews consists of blocks of off days and days they are on call.

Since a portion of open-time flying may be known several days before the day of operations, the crew scheduling department at Jazz Aviation takes such unassigned flights and attempts to cover them using reserve crews before operations day. Approaching the crew rescheduling problem after initial crew schedules are published and before the day of operations allows for a significant portion of open-time flying to be dealt with in a less time-restrained planning environment by utilizing crews as efficiently as possible. In addition, unlike day-of rescheduling, this aspect of crew recovery allows flight crews to know their schedules in advance, which results in more satisfaction. All other open-time flights arising on the day of operations can be dealt with separately. Review of previous studies shows that a gap exists in research between the monthly problem and the day of operations airline crew scheduling problem.

This research addresses the crew pairing problem in the context of open-time flying and reserve crews over a multi-day period that begin operations on the day following scheduling day. For example, the scheduling problem approached on the 20^{th} day of the month deals with open-time flights that would be operating on the 21^{st}, 22^{nd}, 23^{rd} as well as the 24^{th} day of the month. As one of the reasons leading to open-time flying is sudden unavailability of crews, for instance because of sickness, the crew pairing problem will not be merged with the crew assignment problem. Therefore, the objective of this study is to optimally construct legal anonymous pairings respecting the collective bargaining agreement between the airline and the union of flight attendants without considering the needs or preferences of individual crew members.

Crew pairing problems, in the context of airline scheduling, are usually formulated as set partitioning problems (SPP) (Schaefer et al. [30]) or set covering problems (SCP) (Yan & Tu [36]). The set partitioning model does not allow "over-covering of flights", i.e., flights with more crew on board than the number that are required to work (Medard & Sawhney [28]). In theory, the set partitioning model needs the set of all feasible pairings as input (Ball & Roberts, [9]). As there may be billions of legal [cockpit] crew pairings for larger fleets (Schaefer et al. [30]), in practice, a large set of "good" pairings is used (Ball & Roberts, [9]). (Note that airlines typically evaluate crew pairings based on their corresponding planned costs (Schaefer et al. [30])). The solution to the SPP, being a zero-one integer programming problem, is a subset of the pairings, which covers each flight segment exactly once for the scheduling period and at the same time, minimizes the total cost of pairings. The set partitioning problem formulation with regards to the crew pairing problem is as follows:

Parameters:

m: number of flight legs

n: number of legal pairings

c_j: cost of pairing j

$$a_{ij} = \begin{cases} 1, & \text{if flight leg } i \text{ is used on pairing } j \\ 0, & \text{otherwise} \end{cases}$$

Decision Variables:

$$x_j = \begin{cases} 1, & \text{if pairing } j \text{ is used in the solution} \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Min} \sum_{j=1}^{n} c_j \cdot x_j \tag{2.1}$$

subject to:

$$\sum_{j=1}^{n} a_{ij} \cdot x_j = 1, \qquad \forall i \in \{1, \ldots, m\} \tag{2.2}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in \{1, \ldots, n\} \tag{2.3}$$

The objective function (Equation 2.1) minimizes the cost of the selected pairings. Constraint 2.2 ensures that each leg appears only once in a pairing.

Baker et al. [7] provided a general insight as to how the set partitioning formulation of the airline crew scheduling problem must be approached. The solution procedure consists of three stages: enumeration, reduction, and selection. At the enumeration stage, all feasible pairings, which make up the constraint matrix of the set partitioning formulation are generated. The reduction stage is applied within the enumeration process, or afterwards, in order to reduce the problem size and thus the total time required to obtain a solution. Attempts to reduce the problem size, as mentioned by Baker et al. [7], may include reduction procedures such as dominance and logical comparisons, or for instance, elimination of all pairings with layovers at an undesirable base. Baker et al. [7] applied heuristic procedures similar to those used in vehicle routing and scheduling problems to obtain near-optimal solutions for a problem at Federal Express (FedEx). This problem considered 1,000 flights over the course of one month.

Vance et al. [34] presented a two-stage method that does not use the conventional set partitioning model in scheduling pilots. In the first stage, a set of duty periods that cover the flights in the schedule is identified. In the second stage, pairings are built using those duty periods. They mentioned that compared to the traditional SPP formulation, the linear programming (LP) relaxation of their duty-period-based formulation provides a tighter bound on the optimal integer programming (IP) solution, but the disadvantage is that the LP relaxation of their formulation is difficult to solve. Interestingly, the main focus has been put on the LP relaxation of their proposed formulation when attempting to solve the model. Vance et al. [34] have also introduced a modified formulation in order to speed up the convergence of the column generation algorithm. Using the proposed formulations and solution methods, test problems provided by a major American carrier for a domestic daily problem have been modeled and solved. They observed that for a relatively small problem, it is more efficient to enumerate the pairings than to use column generation. The results of the aforementioned research are not known to be implemented and used at any airline.

By examining the results of previous research, Yan and Tu [36] concluded that in relatively large problems, the column generation method does not usually produce an optimal integer solution. Therefore, non-integer solutions obtained by using column generation require further methods for improvement. In addition, extended flight networks cause set partitioning or set covering problems to become more complicated and thus more difficult to solve (Yan & Tu [36]). Yan and Tu [36] formulated and solved the CSP for the cabin crew of a Taiwan airline by constructing a minimum-cost pure network flow model. In their paper, Yan and Tu [36] proposed pure network models to formulate the crew scheduling problem for their specific case study which involves an airline with relatively simple work rules. To be specific, they construct pairings that satisfy dispatch regulations by building a minimum-cost pure network flow model. As Yan & Tu [36] have mentioned, the practical advantage of pure network flow models is that they are very efficient to solve, for example, by using the Network Simplex Method.

A general rule which is included in the Taiwan model of Yan and Tu [36] is that the number of both take-offs and landings in a duty should not be more than four. In other words, each duty period consists of either one or two flight segments. Note that this rule verifies that the flight legs in their study are international flights. On the other hand, restrictions on flying hours and working hours do not allow more than 14 working hours and more than 9 flying hours in any 24-hour time window. Although somewhat similar general restrictions such as a limited number of landings in duty, duration of flying hours and working hours are common in the airline industry, international and domestic scheduling problems differ in details. For instance, in an airline that mostly operates domestic flights, each duty period typically contains more than two flight segments, which means complicated procedures are required to identify legal duty periods.

It can be learned from Yan and Tu's [36] listings of general and work duty constraints that when approaching the crew pairing problem, it is necessary to distinguish between the constraints that must be included in the model and the ones that can either be considered in crew rostering or real-time crew operations. The following constraint of Taiwan airlines makes this statement more clear: *"The number of working hours in a duty can be lengthened if there are incidents. However, adjustments for the irregularity should be reported to the labour union within 24 hours after the event. Suitable rest periods should be offered for overtime after the duty period."* (Yan & Tu [36]) Such a constraint does not appear in the crew pairing model. Rather, it is one that must be dealt with in real-time. This example shows the necessity of understanding which terms in a collective agreement must be included in the pairing construction problem and which ones must be excluded. In addition, it can be concluded that even automating the whole crew scheduling process cannot eliminate the need for crew schedulers in the office.

"Although approaches other than set partitioning have sometimes been successful especially on small instances or under less complicated regulations, SP-based approaches are typically used because they enable the embedding of many complex rules within the variable definition" (AhmadBeygi et al. [3]). AhmadBeygi et al. [3] proposed a mixed-integer programming model to generate pairings that can be solved using commercial solvers to provide academic researchers with a way to easily implement a crew pairing generator in order to test new ideas in the airline planning field. Their modeling approach is based on a set of binary decision variables that determine whether two flights immediately follow each other and, if so, whether they follow each other in the same duty or span the overnight between two duties. They focused on the daily problem, assuming that each flight operates every day of the week and deadheading is not used. In real life situations, some airlines allow deadheading in their models, which is the term used when crews take up revenue seats to reposition for a leg or return to home base following a leg (Arabeyre et al. [6]; Vance et al. [34]) "either by ground or air transportation" (Stojkovic et al. [33]). Deadheading is seen as unproductive flying; however, it can lead to reduced crew costs if used economically (Ball & Roberts [9]). In some cases, especially in long-haul crew pairing problems regarding international flights, the use of deadheads lowers overall costs by eliminating extended rest periods (Barnhart, Hatay & Johnson [10]). AhmadBeygi et al. [3] have mentioned that their formulation takes into account only the most common rules concerning the legality of crew pairings. Therefore, not all obtained valid solutions to their model are legal pairings according to US Federal Aviation Administration (FAA) regulations.

As airline requirements vary substantially from one airline to another, existing modeling techniques for the crew pairing problem could not be adapted directly to the open-time problem. Therefore, a two-phase approach to the open-time pairing optimization problem for reserve cabin crew is proposed, which will be described in Chapter 3.

Chapter 3

Methodology

In this chapter, the two-phase approach to optimize open-time pairings for reserve crew at Jazz Aviation LP, is presented. Phase 1, or the *Feasible Pairing Generation* stage, is explained in Section 3.1. Phase 2, dealing with *Pairing Optimization*, is described in Section 3.2.

3.1 Phase 1: Feasible Pairing Generation

The construction of legal or valid pairings is carried out in Phase 1. Note that the terms *feasible*, *legal* and *valid* are used interchangeably throughout this thesis with regards to crew pairings. Figure 3.1 summarizes valid pairing generation (Phase 1). In Phase 1, when generating pairings from a list of open legs and available deadheads for cabin crew, the following operational characteristics specific to our partner have been taken into account:

1. All flight attendants are qualified to work on all aircraft types, thus fleet types have no impact on the pairing generation problem and are therefore ignored. Note that this may not be true of every airline.

2. In the airline under study, existing aircrafts require 1 or 2 cabin crew positions (purser PU or flight attendant FA) depending on fleet type. Depending on why a segment goes in open-time, one or more crew positions may be affected. Therefore, to generate anonymous pairings that can feasibly be covered by only one crew member, every combination of open flight segment and crew position is given a unique flight identification number, or simply flight ID. It is worth noting that this numbering system is consistent with the leg covering constraint used in the optimization model developed for Phase 2.

Before describing the steps of this phase, it is important to understand the different restrictions that are encountered when building a sequence of flights that can feasibly be flown by cabin crews. This section is organized as follows. Pairing construction constraints are given in Section 3.1.1, where a preprocessing procedure is proposed. Valid pairing generation taking advantage of preprocessing is further developed. Section 3.1.2 deals with the identification of duty periods using *Leg-Gap patterns*. Determination of duty and pairing types is covered in Section 3.1.3 and pairing validation is presented in Section 3.1.4. Finally, pairing costs are examined in section 3.1.5.

3.1.1 Pairing Building Constraints

Constraints that are required to be satisfied when building valid pairings generally fall into two major groups: (a) logical constraints, and (b) airline-specific constraints.

Logical Constraints and Phase 1 Preprocessing

Table 3.1 shows a list of location and time related constraints that must logically/inherently be satisfied by any valid pairing.

In the first phase of the proposed two-phase approach to open-time pairing optimization for reserve cabin crew, firstly the logical constraints presented in Table 3.1 are evaluated with respect to all open segments to decide which legs can potentially follow one another in a pairing. Constraints derived from the airline's collective bargaining

Constraint No.	Description
1	Two segments may follow one another in a pairing (and/or duty period) only if the destination of the first leg is the same as the origin of the next leg.
2	Two segments may follow one another in a pairing (and/or duty period) only if the first leg arrives before the departure time of the next leg while allowing sufficient time for crew to transition between flights.

Table 3.1: Logical constraints on pairing construction

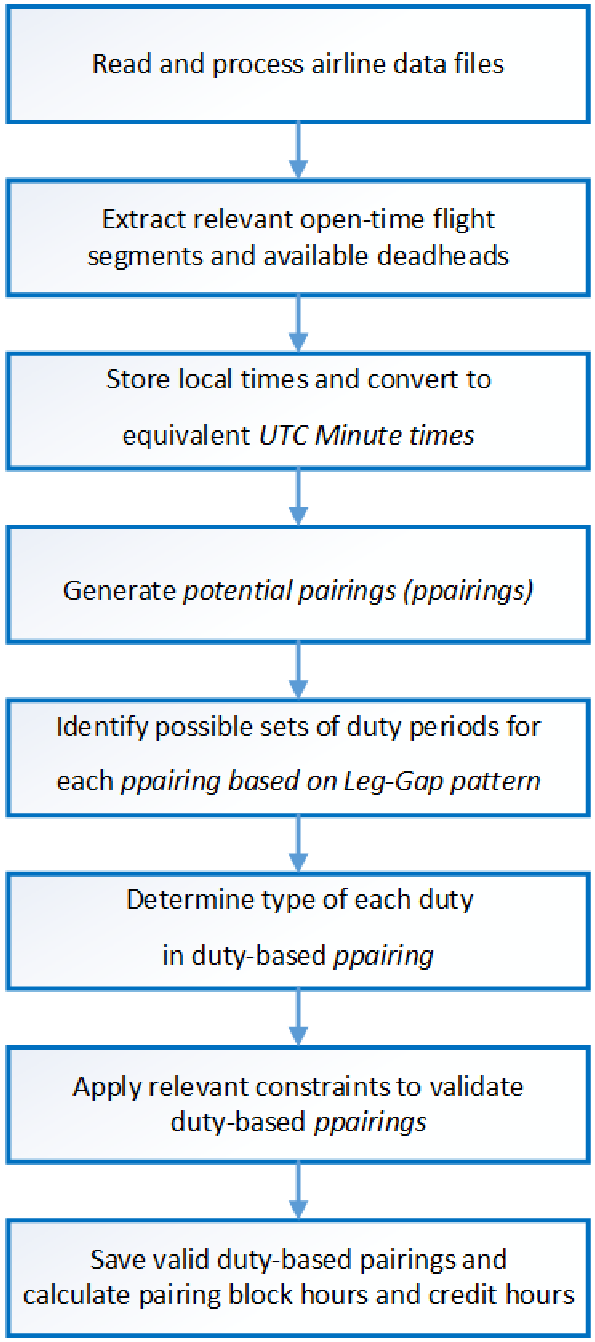


Figure 3.1: Summary of Phase 1 or the Feasible Pairing Generation Phase

agreement (CBA) with the union representing flight attendants are considered afterwards. This approach is based on the idea that regardless of how valid pairings are constructed, a pairing, by definition, as described in Section 1.2.1, is a series of segments. It is clear that in such a series, the idle times between every two consecutive segments (i.e., the gap between one arrival and the next departure) mark sit connections within duty periods and if applicable, rest periods between duties. Therefore, it is rational to begin the valid pairing construction procedure by determining sequences of legs that respect the aforementioned location/time constraints. Such sequences, which will be referred to as *potential pairings* (or *ppairings*) in this text, must contain at least 2 legs due to the base-to-base requirement of pairings and may have up to the maximum number of legs permitted within a pairing. Note that parameters such as the maximum number of legs permitted within a duty period (pairing) are dictated by the collective agreement and will be explored in the next subsection.

The value of the preprocessing procedure above can be understood by evaluating how much the problem size is reduced early on in the process of generating all possible valid pairings given a specific number of open segments. This will be explored in Chapter 4, where sample results showing the number of *potential pairings* resulting from the aforementioned preprocessing procedure is given in Table 4.4.

The pseudo-code for generating *potential pairings* given a set of open segments is as follows:

START.
for every base b:
STEP 1: Initialize ppairing.
Note: ppairing is a list where each element is a sequence of legs.
$j \leftarrow -1$
    for every open leg i:
        if originCity[leg[i]]==base b:
            $j \leftarrow j + 1$

```
            create new ppairing sequence j
            ppairing[j] ←{leg[i]}

        else:
            for every deadhead k from base b:
                if destinationCity[deadhead[k]]==originCity[leg[i]]
                & arrivalTime[deadhead[k]]+DH-IN-BUFFER <departureTime[leg[i]]:
                    j ← j + 1
                    create new ppairing sequence j
                    ppairing[j] ←{deadhead[k] from base b – leg[i]}
                else:
                    continue
    FirstPPairingInCurrentRound ←0
    LastPPairingInCurrentRound ←j


    i ← 0
```

STEP 2: In each iteration, for every sequence created and initialized in STEP 1, add one leg to the end of the sequence if possible and append resulting sequence to list of potential pairings.

```
    Repeat loop until maximum number of legs permitted within a pairing is reached:
        FirstPPairingInPreviousRound ←FirstPPairingInCurrentRound
        LastPPairingInPreviousRound ←LastPPairingInCurrentRound
        # For every ppairing constructed in STEP 1:
        for i in range(FirstPPairingInPreviousRound,LastPPairingInPreviousRound):
            FirstPPairingInCurrentRound ←LastPPairingInPreviousRound + 1
            LastCity ←destinationCity[last leg in ppairing[i]]
            LastTime ←arrivalTime[last leg in ppairing[i]]
            # Compare LastCity and LastTime with originCity and departureTime of every open leg:
            for every open leg j:
                TimeDifference ←departureTime[leg[j]]-LastTime
                if originCity[leg[j]]==LastCity and TimeDifference ⩾ MIN-SIT:
```

create new ppairing sequence ←{ppairing[i] – leg[j]}

else:

continue

LastPPairingInCurrentRound ←index of last ppairing sequence

STEP 3: By the end of this step, potential pairings that begin and end in base are stored in list of all ppairings for all bases.

create empty list of all ppairings for all bases

$i \leftarrow 0$

for every ppairing i:

if destinationCity[last leg in ppairing[i]]$\neq$ base b:

for every deadhead k' to base b:

if originCity[deadhead[k']]==destinationCity[last leg in ppairing[i]] & departureTime[deadhead[k']] >arrivalTime[last leg in ppairing[i]] +DH-OUT-BUFFER:

append sequence {ppairing[i] – deadhead[k'} to list of all ppairings for all bases

else:

continue

else:

append sequence {ppairing[i]} to list of all potential pairings for all bases

END.

The following example shows the result of implementing the above pseudo-code. Sample *potential pairings* for the small set of open legs given in Table 3.2 can be seen in Table 3.3. Note that in this table, the negative flight identification numbers seen in some of the sequences refer to deadhead segments. For each of the airline's crew bases, all of the *potential pairings* are generated. Each *potential pairing* is a sequence of legs where every two consecutive legs respect the logical time/location constraints. In order to respect the base-to-base requirement for pairings, wherever necessary, deadhead segments are used at the beginning and/or end of each *potential pairing*.

For the given example regarding 4 open legs, the constructed *potential pairings* contain 1 to 4 open legs. For instance, the 4^{th} sequence in Table 3.3, is a *potential pairing* based in YYC (Calgary, AB) that contains 2 of the segments in open-time and does not use deadheading. The 5^{th} sequence given in Table 3.3 is another *potential pairing* generated by the aforementioned procedure, which is based in YVR (Vancouver, BC) and uses two deadhead segments to start and finish the pairing in the same base.

Flight ID	Origin Airport	Dept. Time *UTC Minutes*	Dest. Airport	Arrv. Time *UTC Minutes*	Dept. Date	Local Dept. Time	Local Arrv. Time
1	YYC	1,535	YQU	1,621	Day 0	19:35	21:01
2	YQU	1,655	YYC	1,730	Day 0	21:35	22:50
3	YYC	1,775	YXE	1,851	Day 0	23:35	00:51
4	YXE	2,580	YYC	2,663	Day 1	13:00	14:23

Table 3.2: Departure and arrival information required in the construction of *potential pairings* for a small set of open legs

Airline-specific Constraints on Duty Periods, Rest Periods and Pairings

As mentioned earlier, a lengthy list of rules that are, for the most part, specific to each airline, affect the validity of *potential pairings* as constructed above. These constraints have been extracted from the collective agreement and are used to filter the initial list of *potential pairings* step by step in order to obtain the list of valid pairings. Airline-specific constraints can be grouped into three categories: (a) constraints on duty periods, (b) constraints on rest periods, and (c) constraints on pairings. In what follows, each category will be examined in detail.

In the case of the airline under consideration, constraints on duty periods and pairings are complicated by the fact that, as per the collective agreement of flight attendants, there are several types of duty periods and thus pairings. This is due to a period called *silent hours* which is an interval that begins some hour before/after local midnight ($SilentHoursStartTime$) and ends some hour past local midnight ($SilentHoursEndTime$), during which the frequency of flights is lower than during daytime. For example, the Jazz Aviation collective agreement defines silent hours as

No.	Sequence	Origin	Dept. Time *UTC Minutes*	Dest.	Arrv. Time *UTC Minutes*	Dept. Date	Local Dept. Time	Local Arrv. Time
1	[1, -5493]							
		YYC	1,535	YQU	1,621	Day 0	19:35	21:01
		YQU	2,320	YYC	2,408	Day 1	8:40	10:08
2	[1, 2]							
		YYC	1,535	YQU	1,621	Day 0	19:35	21:01
		YQU	1,655	YYC	1,730	Day 0	21:35	22:50
3	[-2397, 2, 3, -5531]							
		YYC	1,325	YQU	1,424	Day 0	16:05	17:44
		YQU	1,655	YYC	1,730	Day 0	21:35	22:50
		YYC	1,775	YXE	1,851	Day 0	23:35	00:51
		YXE	2,305	YYC	2,388	Day 1	8:25	9:48
4	[3, 4]							
		YYC	1,775	YXE	1,851	Day 0	23:35	00:51
		YXE	2,580	YYC	2,663	Day 1	13:00	14:23
5	[-3035, 1, 2, 3, -6114]							
		YVR	980	YYC	1,059	Day 0	9:20	11:39
		YYC	1,535	YQU	1,621	Day 0	19:35	21:01
		YQU	1,655	YYC	1,730	Day 0	21:35	22:50
		YYC	1,775	YXE	1,851	Day 0	23:35	00:51
		YXE	2,145	YVR	2,280	Day 1	5:45	7:00
...								
1707	[1, 2, 3, 4]							
		YYC	1,535	YQU	1,621	Day 0	19:35	21:01
		YQU	1,655	YYC	1,730	Day 0	21:35	22:50
		YYC	1,775	YXE	1,851	Day 0	23:35	00:51
		YXE	2,580	YYC	2,663	Day 1	13:00	14:23

Table 3.3: Sample *potential pairings* generated for the set of open legs given in Table 3.2

the hours between 23:00 p.m. and 05:00 a.m. local time. A given pairing, depending on pairing check-in time and the portion of pairing that falls between silent hours as well as the number of duties it contains may be one of four types, which will be referred to as Pairing Type 1-A, Pairing Type 1-B, Pairing Type 1-C and Pairing Type 2 in this thesis. The three Type 1 pairings are single-duty pairings while the fourth type is multi-duty, meaning that pairings of Type 2 span several days. In addition, based on the local time when a duty period begins, it may either be a Duty Type 1 or a Duty Type 2. Duty Type 1 is an early check-in duty, which means duty check-in time falls between silent hours. If a duty is not Type 1, then it is considered Duty Type 2 (i.e., regular duty). Note that duty check-in time is the time a crew member begins duty and this is not the same as the departure time of the first leg in duty; rather, it is calculated as follows, where CHECK-IN-TIME is a parameter obtained from the collective agreement:

$$\text{DutyCheckInTime} = \text{departureTime[first leg in duty]} - \text{CHECK-IN-TIME} \tag{3.1}$$

A duty period beginning with a leg that departs at 5:55 a.m. local time where flight attendant check-in (CHECK-IN-TIME) is 60 minutes prior to departure, suggests a 4:55 a.m. local check-in which is during silent hours. This is an example of Duty Type 1.

Before listing the specific constraints that affect different types of duty periods and pairings as well as rest periods, it is worthwhile to introduce a few attributes with regards to duty periods and pairings, including duty check-out time, pairing check-in time, pairing check-out time, length of duty period, length of pairing, length of rest period, number of landings within duty, and number of duties within pairing.

Duty check-out time marks the end of a duty period and must be distinguished from the arrival time of the last leg in duty:

$$\text{DutyCheckOutTime} = \text{arrivalTime[last leg in duty]} + \text{CHECK-OUT-TIME} \quad (3.2)$$

In the above formula, similar to CHECK-IN-TIME, CHECK-OUT-TIME is a parameter obtained from the collective agreement. Note that in the simplest of cases, both parameters may have constant values. For instance, in an airline, flight attendants may be required to check in 60 minutes prior to the first departure in duty and to check out 30 minutes after the last arrival. However, this is not always the case in practice. As a matter of fact, where trans-border flights are present in the flight schedule in addition to domestic legs, the amount of check-out time required following the arrival of the last segment depends on whether or not customs need to be cleared. If customs must be cleared, a longer CHECK-OUT-TIME becomes essential in the pairing construction problem.

Check-in and check-out times for a pairing are also of importance and can be calculated in a similar fashion as duty periods:

$$\text{PairingCheckInTime} = \text{departureTime[first leg in pairing]} - \text{CHECK-IN-TIME} \quad (3.3)$$

$$\text{PairingCheckOutTime} = \text{arrivalTime[last leg in pairing]} + \text{CHECK-OUT-TIME} \quad (3.4)$$

The length of a duty period is the elapsed time between duty check-in and duty check-out times, which is referred to as Duty-Elapsed-Time in this thesis. According to flight crew work rules, duty length is not to exceed a maximum limit (MAX-LENGTH-OF-DUTY), which varies by type of duty and pairing.

$$\text{Duty-Elapsed-Time} = \text{DutyCheckOutTime} - \text{DutyCheckInTime} \quad (3.5)$$

In addition, the length of the time interval between check-in and check-out times of a pairing is referred to as Pairing-Elapsed-Time.

$$\text{Pairing-Elapsed-Time} = \text{PairingCheckOutTime} - \text{PairingCheckInTime} \qquad (3.6)$$

Note that the subtraction operations above must be carried out carefully using appropriate functions due to the operands being time objects that can occur in different time zones.

For a pairing that spans multiple work duties, namely Pairing Type 2, between every two consecutive duty periods, a rest period is required, the length of which must satisfy a minimum amount (MIN-LENGTH-OF-LAYOVER) as crew members need to rest after every work day. It is worth noting that the aforementioned lengths of time will later appear in cost calculations.

Another attribute worth mentioning here is the number of landings within a duty. Based on work and safety regulations, each duty period within a pairing may consist of at most a specific number of flight segments. Note that the term *landing* is usually used in collective agreements in this regard in order to distinguish between similar flight numbers that make at least one stop before landing in the final destination. For example, flight number 7063 departing from YAM at 11:50 a.m. local time and arriving in YWG at 13:50 p.m. local time needs to be distinguished from flight number 7063 leaving YWG at 14:35 p.m. local time and landing in YYC at 16:40 p.m. local time. For pairing building purposes, this flight number marks two segments, each of which, as explained before, is basically a non-stop flight. Therefore, the presence of both portions of this flight number in a duty period is equivalent to two landings.

In the simplest case, the maximum number of legs permitted within a duty period (MAX-NUMBER-OF-LEGS-IN-DUTY) is a constant parameter. However, in the case where there are several types of duty periods and pairings, this number varies by type. In fact, at the airline under consideration, in one case, this maximum value is variable, meaning that depending if a certain amount of break is provided during duty, the number of segments may be increased as well as the length of duty. In addition, deadheading affects the permitted number of segments in duty, meaning that for instance, if a deadhead segment is present in a duty period of Type 2, an extra segment is allowed. As it can be seen, various factors contribute to determining parameters such as maximum landings in duty and maximum length of duty, which cannot be known in advance. The approach proposed in this thesis requires that *potential pairings* first be broken down to possible duty periods, and then the type of every resulting *duty-based pairing* be identified. It is when the type of each *potential pairing* and its constituting duty periods are specified that the appropriate constraints on duty periods, rest periods and pairings can be applied to filter out invalid *potential pairings*.

Similar to the number of segments in a duty, the maximum number of duty periods permitted within a pairing is limited. A single-day pairing consists of one duty period, whereas a multi-day pairing is made up of several duties separated by periods of rest, in which the number of duties is not to exceed MAX-NUMBER-OF-DUTIES -a parameter obtainable from the collective agreement that typically has a value of 4 or 5 for cabin crew.

As mentioned before, rest periods or layovers happen between every two consecutive duty periods in a multi-day pairing. Depending on the airline, there may be a restriction as to where, geographically speaking, layovers take place as the airline is generally required to book hotel rooms for crew to rest before their next work day. Understanding company policies is especially important in airlines with multiple crew bases regarding layovers at or outside home base. For instance, in the developed open-time pairing optimization tool for reserve flight attendants, as per the airline's request, overnight at home base does not appear in any pairing.

Tables 3.4, 3.5, and 3.6 list constraints on duty periods, rest periods and pairings for the case of the industry partner.

Constraint No.	Description
3	Number of landings within a duty period must not exceed maximum permitted (MAX-NUMBER-OF-LEGS-IN-DUTY). **Complexity:** Parameter MAX-NUMBER-OF-LEGS-IN-DUTY varies by type of duty/pairing and whether or not there is a deadhead in duty.
4	Length of a duty period must not exceed maximum permitted (MAX-LENGTH-OF-DUTY). **Complexity:** Parameter MAX-LENGTH-OF-DUTY varies by type of duty/pairing.

Table 3.4: Airline-specific constraints regarding duty periods

Constraint No.	Description
5	Length of rest period between every two consecutive duty periods within a multi-day pairing must satisfy a minimum amount (MIN-LENGTH-OF-LAYOVER).
6	Multi-day crew pairings may not overnight at home base.

Table 3.5: Airline-specific constraints regarding rest periods or layovers

3.1.2 Identifying Duty Periods

In order to apply the listed constraints to the previously-constructed *potential pairings*, due to the fact that parameters such as MAX-LENGTH-OF-DUTY and MAX-NUMBER-OF-LEGS-IN-DUTY depend on duty type, it is first necessary to identify duty periods within each *potential pairing*. In what follows, the duty period identification procedure is described in detail.

As defined earlier, a *potential pairing*, which meets the minimum requirements of a pairing (logical constraints), is a sequence of legs. Since flight legs each have a known departure time and arrival time, there is naturally a time gap between every leg and its following segment in the pairing. The length of this gap is the length of time between an arrival and the departure time of the next leg. Given a *potential pairing*, using the departure and arrival times of its legs, it is possible to calculate the lengths of all such gaps. These calculations are very simple to carry out due to the fact that the times that need to be subtracted from one another are in the same time

Constraint No.	Description
7	Number of duty periods within a pairing cannot exceed the maximum limit (MAX-NUMBER-OF-DUTIES).

Table 3.6: Airline-specific constraints regarding pairings

zone. In the process of identifying duty periods which in turn determines layovers, what is of interest is the gaps in a *potential pairing* which are at least as long as MIN-LENGTH-OF-LAYOVER. This process is based on the fact that in a *potential pairing*, only gaps that are sufficiently long have the potential to mark rest periods in the pairing. Legs that fall between periods of rest constitute duty periods. In this thesis, the term *Gap* refers to the described period of time.

Depending on the number of *Gaps* available in a *potential pairing*, several combinations of duty periods may be possible. After applying constraints derived from work rules, some of these combinations may be found infeasible. However, there is always one case that is more likely to be feasible: all available *Gaps* are considered layovers that separate duty periods. Before moving on to the relationship between the number of identified *Gaps* and the number of possible combinations of duty periods in a *potential pairing*, *Leg-Gap* patterns will be introduced.

Every *potential pairing* can be described using a pattern which will be referred to as *Leg-Gap* pattern in this thesis and is employed in the duty identification process. The idea of associating *Leg-Gap* patterns to pairings helps design an algorithm that can break any pairing down to duty periods, which, as mentioned earlier, is a necessary step before a *potential pairing* can be validated. Given a *potential pairing*, each unique pattern shows how flight legs and identified *Gaps* follow one another. For example, the *Leg-Gap* pattern associated with the following *potential pairing* of legs with identification numbers 10, 3, 4, 7, 16, and 28 in the given order, with the time gap between the arrival time of leg number 4 and the departure time of leg number 7 being greater than or equal to MIN-LENGTH-OF-LAYOVER (e.g., 10 hours, or 600 minutes), is *L-L-L-G-L-L-L*. This is an example of a *potential pairing* with 1 *Gap*. An example of the *Leg-Gap* pattern for a *potential pairing* with 3 *Gaps*, which may in turn have 3 layovers and thus 4 duty periods, is *L-L-L-L-G-L-L-L-L-G-L-L-L-G-L-L-L*. In what follows, the impact of the number of *Gaps* in a *Leg-Gap* pattern on how pairings can be broken down to duty periods will be explored.

The only factor that plays a role in determining the number of possible combinations of duty periods for a given *potential pairing* is the number of *Gaps*. Table 3.7 helps illustrate this idea. In this table, sample *potential pairing Leg-Gap* patterns as well as all probable duty periods associated with each pattern are given. Note that for each sample *potential pairing*, the case which is most likely feasible is marked with a *. Since the number of duties in a pairing is limited to MAX-NUMBER-OF-DUTIES (for instance, 4), only patterns that may result in the acceptable number of duties and layovers need to be inspected. For instance, assuming the maximum number of duty periods permitted within a pairing is 4, a pattern that contains 8 *Gaps* cannot result in a feasible pairing and thus does not require further processing. The logic behind the duty period identification procedure, which will be explored shortly, sheds light on this argument.

Table 3.7: Relationship between number of Gaps (G) in *potential pairing* pattern and number of possible sets of duty periods

Example	
PPairing Pattern	Possible Duty Periods
L-L-L	*Case 1: [Duty # 1 of 1: L-L-L]
L-L-G-L-L	*Case 1: [Duty # 1 of 2: L-L]-[layover]-[Duty # 2 of 2: L-L]
	Case 2: [Duty # 1 of 1: L-L-[long sit connection]-L-L]
L-L-G-L-L-G-L-L	*Case 1: [Duty # 1 of 3: L-L]-[layover]-[Duty # 2 of 3: L-L]-[layover]-[Duty # 3 of 3: L-L]
	Case 2: [Duty # 1 of 2: L-L]-[layover]-[Duty # 2 of 2: L-L-[long sit connection]-L-L]
	Case 3: [Duty # 1 of 2: L-L-[long sit connection]-L-L]-[layover]-[Duty # 2 of 2: L-L]

Table 3.7 Continued

PPairing Pattern	Possible Duty Periods
L-L-G-L-L-G-L-L-G-L-L	*Case 1: [Duty # 1 of 4: L-L]-[layover]-[Duty # 2 of 4: L-L]-[layover]-[Duty # 3 of 4: L-L]-[layover]-[Duty # 4 of 4: L-L]
	Case 2: [Duty # 1 of 3: L-L]-[layover]-[Duty # 2 of 3: L-L]-[layover]-[Duty # 3 of 3: L-L-[long sit connection]-L-L]
	Case 3: [Duty # 1 of 3: L-L]-[layover]-[Duty # 2 of 3: L-L-[long sit connection]-L-L]-[layover]-[Duty # 3 of 3: L-L]
	Case 4: [Duty # 1 of 3: L-L-[long sit connection]-L-L]-[layover]-[Duty # 2 of 3: L-L]-[layover]-[Duty # 3 of 3: L-L]
	Case 5: [Duty # 1 of 2: L-L-[long sit connection]-L-L]-[layover]-[Duty # 2 of 2: L-L-[long sit connection]-L-L]

To understand how potential duty periods are identified for a given *potential pairing*, consider an example with two *Gaps*. In Table 3.7, the sequence *L-L-G-L-L-G-L-L* implies there is enough time between the arrival of the second leg and the departure of the third leg to allow for a layover if needed. This is also true of the fourth arrival and the fifth departure. If the entire *potential pairing* which covers legs 1 through 6, is not broken into smaller duty periods separated by periods of rest, it is considered a single duty period consisting of 6 segments and 2 long sit connections. In what follows, the feasibility of such a duty period is explored. Note that at this point, the constraints of interest are those relevant to lengths of time regarding duty periods.

Assuming MAX-LENGTH-OF-DUTY is, depending on type of duty, either 12 hours (720 minutes) or 13 hours and 30 minutes (810 minutes), and that MIN-LENGTH-OF-LAYOVER is 10 hours (600 minutes), since 2 *Gaps* add up to at least 20 hours (1200 minutes), two or more *Gaps* are not feasible in the same duty period, regardless of duty type. Therefore, each duty period may contain at most 1 *Gap*. As a result, given the *Leg-Gap* pattern for a *potential pairing*, considering that every *Gap* in the pattern may either be a layover or a long sit connection, and taking into account the fact that at most one *Gap* may exist in a feasible duty period, then based on the number of *Gaps* in the *potential pairing*, several combinations of duty periods are possible. In a pattern containing two *Gaps*, as mentioned above, it is not feasible to consider both *Gaps* to be long sit connections in a single duty period. Therefore, the search for the duty periods in *L-L-G-L-L-G-L-L*, as seen in Table 3.7, leads to three different combinations which will be referred to as *duty-based pairings* in this thesis:

1: (1^{st} *Gap*: layover , 2^{nd} *Gap*: layover) $\rightarrow$ 3 duty periods D_1, D_2, D_3

2: (1^{st} *Gap*: layover , 2^{nd} *Gap*: long sit connection) $\rightarrow$ 2 duty periods D'_1, D'_2

3: (1^{st} *Gap*: long sit connection , 2^{nd} *Gap*: layover) $\rightarrow$ 2 duty periods D''_1, D''_2

It is worth mentioning that the likelihood of each of the above cases being found valid eventually is different. In fact, a rough comparison of MAX-LENGTH-OF-DUTY, MIN-LENGTH-OF-LAYOVER and the typical length of flights considered shows that the case where all *Gaps* in the *potential pairing* are assumed to be rest periods, such as *Case 1* listed above, has a higher probability of being feasible than the other cases where one or more duty periods include a long sit connection in addition to flight legs. However, in such cases as *Case 2* and *Case 3* above, there is a chance, although small, that each resulting *duty-based pairing* is found feasible when the remaining constraints are applied.

Based on the described logic, the developed decision support system for open-time cabin crew pairing optimization has a function that can take the *Leg-Gap* pattern associated with a *potential pairing* to produce *duty-based pairings*, which are the sequences that the airline-specific constraints explained earlier can directly be applied to once the type of pairing and its duty periods are determined.

The total number of possible *duty-based pairings* for different number of *Gaps* found in *Leg-Gap* patterns, is given in Table 3.8. These numbers are derived based on the values listed in Table 3.9. As mentioned earlier, a pattern that contains 8 *Gaps* (or more) cannot result in a feasible pairing. To understand why, consider a basic *Leg-Gap* pattern consisting of 8 *Gaps*: *L-G-L-G-L-G-L-G-L-G-L-G-L-G-L-G-L*. In order for the underlying *potential pairing* to have a maximum of 4 duty periods, at most 3 of its 8 *Gaps* must be layovers. Therefore, the remaining 5 *Gaps* must be long sit connections that appear in 4 duty periods, which means that there must be at least 2 *Gaps*/long sit connections in a single duty. This violates the constraint on the maximum permitted length of duty and thus results in the *potential pairing* being infeasible.

3.1.3 Determining Duty and Pairing Types

Following the *potential pairing* duty period identification procedure, for every resulting *duty-based pairing*, the type of each pairing as well as duty period if applicable, is specified. For the airline under study, the specific criteria that define each category of pairings and duty periods can be found in the collective agreement and are utilized to determine whether each pairing is Pairing Type 1-A, 1-B, or 1-C, or Type 2. Note that a *potential pairing* that does not contain any *Gaps* is a single duty period or in other words, a single-day pairing, which is one of the three Type 1 pairings. On the

Number of *Gaps*	Number of Resulting *Duty-based Pairings*
0	1
1	2
2	3
3	5
4	7
5	7
6	4
7	1
8 or more	0

Table 3.8: The relationship between the number of identified *Gaps* in a *Leg-Gap pattern* and the number of possible combinations of duty periods, based on the assumptions provided in Table 3.9

Parameter	Value
MAX-LENGTH-OF-DUTY	12 hours or 13 hours and 30 minutes (depending on type of duty)
MIN-LENGTH-OF-LAYOVER	10 hours
MAX-NUMBER-OF-DUTIES	4

Table 3.9: Assumed values of problem parameters used in the described duty period identification procedure

other hand, multi-day pairings include two or more duties and are referred to as Pairing Type 2. In the case of Type 2 pairings, the question that needs to be answered is whether each of the duty periods within the pairing are Duty Type 1 or 2. The developed decision support system executes this step and saves the pairing/duty types of each *duty-based pairing* for the purpose of applying the appropriate constraints in order to remove invalid *potential pairings*. For confidentiality reasons, in this thesis, the details regarding how each type of pairing or duty is defined will not be discussed.

3.1.4 Pairing Validation

Once the aforementioned steps are carried out and duty and pairing types are achieved, *duty-based pairings* are qualified to be passed to the *potential pairing* validation procedure explained earlier. For each of the previously-constructed *duty-based [potential]* *pairings*, appropriate airline-specific constraints are checked in order to remove invalid (or illegal) pairings. Cycling through the entire list of *duty-based [potential] pairings* results in the list of all valid pairings across all crew bases. References to the constraints that are applied to duty periods, rest periods and overall pairings are found in Tables 3.4, 3.5, and 3.6. The output of the validation code is a binary variable for every existing *duty-based pairing*, which is set to 1 (or YES) if a pairing is found invalid due to at least one of the constraints being violated, or set to 0 (or NO) otherwise. Therefore, only *duty-based pairings* for which the variable PPAIRING-IS-INVALID is equal to 0 are saved to the list of valid pairings. Before discussing the costs associated with each feasible pairing generated in Phase 1 of the proposed approach to open-time crew scheduling and moving on to the pairing optimization phase (Phase 2), an example will be provided to illustrate how the *potential pairing* validation code functions.

Consider the sample *potential pairing* $\{11 - 12 - 17 - 18 - 19 - 20\}$ which has the following *Leg-Gap pattern*: *L-L-G-L-L-L-L*. The duty period identification code produces the following *duty-based pairings* as output:

1. $\{11 - 12 - \textit{layover} - 17 - 18 - 19 - 20\}$

2. $\{11 - 12 - 17 - 18 - 19 - 20\}$

Pairing 1, in the above example, has two duty periods, namely D_1: $\{11 - 12\}$ and D_2: $\{17 - 18 - 19 - 20\}$, while Pairing 2 has a single duty period. The second *duty-based pairing* is known to match the definition of Pairing Type 1-C, for which the length of pairing, or equivalently length of duty, is limited to 13 hours and 30 minutes (i.e., MAX-LENGTH-OF-DUTY=810 minutes). Knowing this, appropriate parameters are substituted in the constraints. Running the *potential pairing* validation procedure finds that this *potential pairing* is invalid. This is because although the constraint on the number of legs within duty is satisfied, it violates the constraint regarding total length of duty. The length of Pairing 2, found using converted *UTC Minutes*, is:

$$\text{Pairing-Elapsed-Time} = \text{PairingCheckOutTime} - \text{PairingCheckInTime} \quad (3.7)$$
$$= 2239 - 1235 = 1004 \text{ minutes} \;>\; \text{permitted 810 minutes}$$

On the other hand, Pairing 1, being composed of multiple duty periods, undergoes a slightly different validation procedure than Pairing 2. Before getting into the details of each duty period which is where most of the restrictions are in place, the overall pairing and rest period(s) are checked. Since the minimum amount of rest required between consecutive duties (i.e., parameter MIN-LENGTH-OF-LAYOVER) is embedded in the proposed approach to identifying duty periods, it is not necessary to assess the length of rest periods in the validation code. However, in the open-time crew pairing problem for Jazz Aviation LP, the cities where layovers take place are to be compared against crew bases as multi-day pairings may not overnight at home base. Pairing 1 begins and terminates at YVR, therefore this restriction leads to the entire pairing being infeasible since the layover follows the arrival of crew at the destination of leg 12, which also happens to be YVR. Logically, it is not necessary to continue evaluating the remaining constraints once the violation of a constraint results in the pairing being found infeasible. However, in this example, for the sole purpose of clarifying how the validation procedure works, the pairing is treated as if it were valid.

As for the constraint on pairings, as explained before, the number of duties within a feasible pairing does not exceed MAX-NUMBER-OF-DUTIES. In the given example, MAX-NUMBER-OF-DUTIES has a value of 4, which means Pairing 1 has an acceptable number of duty periods. At this point, the question that needs to be answered is whether or not each duty within Pairing 1 satisfies the constraints regarding duty periods.

Duty D_1, with a check-in time of 12:35 p.m. local YVR time and check-out time of 19:15 p.m. local YVR time on the same day, is Type 2 Duty (i.e., regular duty as opposed to early check-in). In this example, for Type 2 Duty, MAX-LENGTH-OF-DUTY is 13 hours and 30 minutes (or equivalently, 810 minutes), and MAX-NUMBER-OF-LEGS-IN-DUTY is equal to 7 if there are no deadhead segments in duty or 8 otherwise. The length of duty for D_1 is calculated using converted *UTC Minutes*:

$$\begin{aligned}\text{Duty-Elapsed-Time} &= \text{DutyCheckOutTime} - \text{DutyCheckInTime} \\ &= 3075 - 2675 = 400 \text{ minutes} \ < \ \text{MAX-LENGTH-OF-DUTY}\end{aligned} \tag{3.8}$$

Therefore D_1, which includes 2 open legs and 0 deadheads, does not violate any of the constraints concerning Type 2 duties. The second duty in Pairing 1, which follows a rest period of 1055 minutes (or 17 hours and 35 minutes), is analyzed in a similar way.

For D_2, check-in is at 12:50 p.m. local YVR time and check-out is at 22:39 p.m. local YVR time on the date following the first duty. Similar to D_1, D_2 is Type 2 Duty. The length of duty is calculated using converted *UTC Minutes*:

$$\begin{aligned}\text{Duty-Elapsed-Time} &= \text{DutyCheckOutTime} - \text{DutyCheckInTime} \\ &= 4719 - 4130 = 589 \text{ minutes} \ < \ \text{MAX-LENGTH-OF-DUTY}\end{aligned} \tag{3.9}$$

Thus, D_2, consisting of 4 open segments and 0 deadheads, satisfies the constraints on Type 2 duties and is a valid duty period. However, in this example, the entire pairing, despite having feasible duties, is infeasible for the reason explained earlier and therefore the variable PPAIRING-IS-INVALID corresponding to Pairing 1 is set to 1. Following the validation of *potential pairings*, the costs associated with valid crew pairings must be calculated in order for the optimization model of Phase 2 to select the optimal, i.e., minimum cost, set of feasible pairings.

3.1.5 Pairing Cost Calculations

The specific cost structure of crew pairings varies across airlines. Typically, North American airlines pay their crews based on *credit hours*, which are the units of work that a crew member earns for pay purposes, whereas European airlines often pay crew a fixed salary (Andersson et al. [5]). Depending on crew-pay structure, the objective of the optimization model in the crew pairing problem takes different forms, such as minimization of crew costs or maximization of crew utilization (or equivalently, minimization of the required number of flight crews or minimization of the number of pairings required to cover the flight schedule) (Dück, Wesselmann & Suhl, [19]). In the problem under study, since the partner uses credit hours to pay flight attendants, the objective is introduced as the minimization of total *credit loss*. The goal is to select a collection of legal pairings where the number of paid working hours is as close as possible to the number of actual flying hours, subject to each segment being covered once. In mathematical terms, this objective is defined as the minimization of the ratio of total credit hours over total block hours across all selected eligible pairings, or

$$Objective_1 : minimize \frac{\sum_{i=1}^{n}(CreditHours_i \cdot x_i)}{\sum_{i=1}^{n}(BlockHours_i \cdot x_i)} \tag{3.10}$$

Equation 3.10 is written as defined by the industry partner. It will be linearized for optimization purposes. Before discussing the pairing selection phase in detail, it is important to understand how credit hours and block hours are defined in the collective agreement. Block hours for an individual flight leg, in the context of the cabin crew pairing problem, is the duration of the flight, i.e., the elapsed time between the departure time at the origin and the arrival time at the destination. For each flight in the list of open segments, this parameter is usually available in $hh : mm$ format. Where several flight segments are sequenced in a duty period and/or pairing, block hours is defined as the total duration of productive flying time. Thus, for a duty period, block hours (*Duty-Block-Hours*) is calculated as the sum of the individual block hours for each flight segment in the duty. However, it must be noted that if a deadhead flight is present in a duty period, the associated block hours is not used in the calculations since, by definition, deadheads are only used to transport crew as passengers in order to begin working elsewhere and thus are not considered productive flying hours. In a pairing composed of several duty periods, block hours (*Pairing-Block-Hours*) can be calculated as the sum of the block hours for each duty in the pairing. *Duty-Block-Hours* and *Pairing-Block-Hours*, which are used in finding credit hours, are calculated as follows:

$$Duty\text{-}Block\text{-}Hours = \sum_{\substack{k \in \{1,\ldots,L\} \\ k \notin Deadheads}} BlockHours\{Leg[k]\} \tag{3.11}$$

$$Pairing\text{-}Block\text{-}Hours = \sum_{d \in \{1,\ldots,D\}} BlockHours\{Duty[d]\} \tag{3.12}$$

Credit hours has a more complicated structure compared to block hours for both duty periods and pairings. Although the specific factors that impact crew pay on the basis of credit hours may vary from airline to airline, credit hours is generally a complex non-linear function of several parameters such as minimum guaranteed pay per duty, actual flying hours (i.e., block hours) and time away from base or TAFB (i.e., the number of minutes that elapse between the beginning and the end of a duty period/pairing (Schaefer et al. [30]). For instance, the developed open-time pairing generator and optimizer for cabin crew uses the *maximum* function to determine credit hours for duty periods (*Duty-Credit-Hours*) as well as pairings (*Pairing-Credit-Hours*). *Duty-Credit-Hours* is the maximum of three values and can be found using the following function:

$$Duty\text{-}Credit\text{-}Hours = \max \begin{cases} \text{MIN-GUARANTEED-CREDIT-HRS-PER-DUTY;} \\ Duty\text{-}Block\text{-}Hours; \\ w_1 \cdot Duty\text{-}Elapsed\text{-}Time. \end{cases} \tag{3.13}$$

In the above equation, MIN-GUARANTEED-CREDIT-HRS-PER-DUTY represents the minimum number of credit hours crew are guaranteed to get paid per duty period regardless of the length of duty, and is a constant parameter specified in the collective agreement. *Duty-Elapsed-Time* is calculated as the length of the time interval that begins at DutyCheckInTime and ends at DutyCheckOutTime (given in Chapter 3).

Parameter w_1 is a constant derived from CBA with a fractional value between 0 and 1 ($0 < w_1 < 1$). This parameter represents the significance weight given to the length of duty periods regarding credit hours. The term $w_1 \cdot \textit{Duty-Elapsed-Time}$ in Equation 3.13 indirectly incorporates the cost of deadheading on credit hours. For a multi-day pairing, once *Duty-Credit-Hours* are calculated for each duty period, *Pairing-Credit-Hours* can be determined as follows:

$$\textit{Pairing-Credit-Hours} = \max \begin{cases} \displaystyle\sum_{d \in \{1,\ldots,D\}} \textit{Duty-Credit-Hours}\{\textit{Duty}[d]\}; \\ w_2 \cdot \textit{Pairing-Elapsed-Time}. \end{cases} \tag{3.14}$$

In this equation, *Pairing-Elapsed-Time*, commonly referred to as time away from base or TAFB, is the length of the time interval that begins at PairingCheckInTime and ends at PairingCheckOutTime (given in Chapter 3), and w_2 is a constant parameter with a predefined fractional value between 0 and 1 ($0 < w_2 < 1$). Similar to w_1, w_2 in Equation 3.14 indirectly incorporates the impact of deadheading on credit hours. It is worth noting that $w_1 > w_2$. As a result, finding the credit hours for single-duty pairings reduces to only calculating *Duty-Credit-Hours*. The calculation of cost parameters for valid pairings, namely *Pairing-Block-Hours* and *Pairing-Credit-Hours*, marks the end of the first phase. All valid legal pairings and their costs are recorded in specially formatted matrices that are passed to the next phase.

3.2 Phase 2: Pairing Optimization

Phase 2, or the Pairing Optimization or Selection Phase, uses the output of Phase 1 as input data in running the BIP formulation to be developed to select the optimal pairings covering the segments in open time.

Using the matrices generated in Phase 1, the Pairing Selection Problem is formulated as a Set Partitioning Problem (SPP), with the exception that the objective function is modified to incorporate deadheading preferences. This formulation is given in Equations 3.15 through 3.17. Note that to find the value of parameter n' used in these equations, the number of open legs is added to the total number of valid duty-based pairings generated in Phase 1. The reason behind this will be discussed in Section 4.5.

Parameters:

m: number of flight legs

n': number of legal pairings

C_j: PAIRING-CREDIT-HOURS for pairing j

B_j: PAIRING-BLOCK-HOURS for pairing j

L_j: Pairing-Elapsed-Time for pairing j

d: very small positive number. Here $d = 0.000001$.

$$A_{ij} = \begin{cases} 1, & \text{if open flight leg } i \text{ is used on pairing } j \\ 0, & \text{otherwise} \end{cases}$$

Decision Variables:

$$x_j = \begin{cases} 1, & \text{if pairing } j \text{ is used in the solution} \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Min} \sum_{j=1}^{n'} (C_j - B_j + d \cdot L_j)\, x_j \tag{3.15}$$

s.t.

$$\sum_{j=1}^{n'} A_{ij}\, x_j = 1, \qquad \forall i \in \{1, \ldots, m\} \tag{3.16}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in \{1, \ldots, n'\} \tag{3.17}$$

The objective function (Equation 3.15) has two terms. The first term is a linearized version of the credit loss ratio introduced in Equation 3.10: Total credit loss is expressed as the difference between total credit hours and total block hours across all legal pairings.

The second term of the objective function aims to capture the preference given to shorter deadheads. The specific cost structure that applies to duty periods and pairings, which was described previously, may lead to two or more legal pairings with similar values for PAIRING-CREDIT-HOURS – PAIRING-BLOCK-HOURS. This occurs in pairings that consist of the same open legs but only differ in the deadhead segments that may have been used in order to fly crew from and/or to any of the airline's several bases at the beginning and/or end of pairings. In such cases, the pairing with the lowest value for Pairing-Elapsed-Time would be equivalent to the pairing with the least deadheading. For economical, operational and safety reasons, airlines prefer to have crew spend less time on deadheads. Therefore, the length of pairings is used to select the most desirable pairing among pairings with similar credit loss. It is worth noting that a very small weight, $d = 0.000001$ or 10^{-6}, has been arbitrarily chosen to be applied to the pairing lengths in order to distinguish between the main objective of this minimization and what is only a tiebreaker.

Equation 3.16 is the leg covering constraint, which states that each open leg must appear in one and only one of the selected pairings. Feasible pairings, generated in Phase 1, are composed of open-time flight segments and possibly deadhead legs at the beginning and/or end of pairings; however, since covering deadhead legs is not the purpose of the optimization problem, they are excluded from the constraint matrix $[A_{ij}]_{m \times n'}$, where the rows represent unique flight ID of open legs and the columns show legal pairings.

The optimal solution found by solving the given mathematical model using available linear/integer programming solvers is an assignment of the values 0 or 1 to the decision variables (Equation 3.17) and is interpreted as follows: All j's for which $x_j = 1$, make up the subset of legal pairings that have been selected by the optimization phase, resulting in the minimum total cost. Referring back to the list of pairings generated and validated in Phase 1, details regarding the legs that fall on each selected pairing j can be retrieved. In order to access a user-friendly summary of the results of the optimization, the developed decision support system includes a post-processing module that displays essential information with regards to the sequence of flights on each pairing, such as flight number, departure date, origin airport, departure time, destination airport and arrival time for deadhead flights and segments that were originally in open-time. Figure 3.2 shows what the output of the developed decision tool typically looks like. Crew schedulers using the pairings generated by this tool would then need to make the necessary arrangements for the flying of crew on specified flights for the purpose of deadheading which may take place on flights operated by other airlines, as well as to book hotel rooms during layovers within pairings.

3.2.1 Extending the Optimization Objective to Reduce Deadheading

Different metrics inspire the extension of the optimization objective beyond cost minimization. For example, the ratio of the total number of used deadhead segments across all of the selected pairings to the number of open legs can be incorporated in the objective function of the optimization problem. Weight parameters assigned to each term in the objective show relative preference among optimization goals. The crew pairing optimization model of Phase 2 can be reformulated as follows (Equations 3.18 to 3.20) assuming that newly-introduced parameters D_j are available for every valid duty-based pairing. (In Phase 1, the developed decision support system can easily calculate and save these parameters in matrix form.)

Parameters:

m: number of open flight legs

n': number of legal pairings

C_j: PAIRING-CREDIT-HOURS for pairing j

B_j: PAIRING-BLOCK-HOURS for pairing j

L_j: Pairing-Elapsed-Time for pairing j

d: very small positive number. Here $d = 0.000001$.

w_3: arbitrary number between 0 and 1

$$D_j = \begin{cases} 0, & \text{if pairing } j \text{ does not have any deadhead legs} \\ 1, & \text{if pairing } j \text{ has one deadhead leg (from or to a base)} \\ 2, & \text{if pairing } j \text{ has two deadhead legs (from a base and to the same base)} \end{cases}$$

$A_{ij} = \begin{cases} 1, & \text{if open flight leg } i \text{ is used on pairing } j \\ 0, & \text{otherwise} \end{cases}$

<u>Decision Variables:</u>

$x_j = \begin{cases} 1, & \text{if pairing } j \text{ is used in the solution} \\ 0, & \text{otherwise} \end{cases}$

$$\text{Min } \sum_{j=1}^{n'} (C_j - B_j)\, x_j + d \sum_{j=1}^{n'} L_j\, x_j + w_3 \frac{1}{m} \sum_{j=1}^{n'} D_j\, x_j \tag{3.18}$$

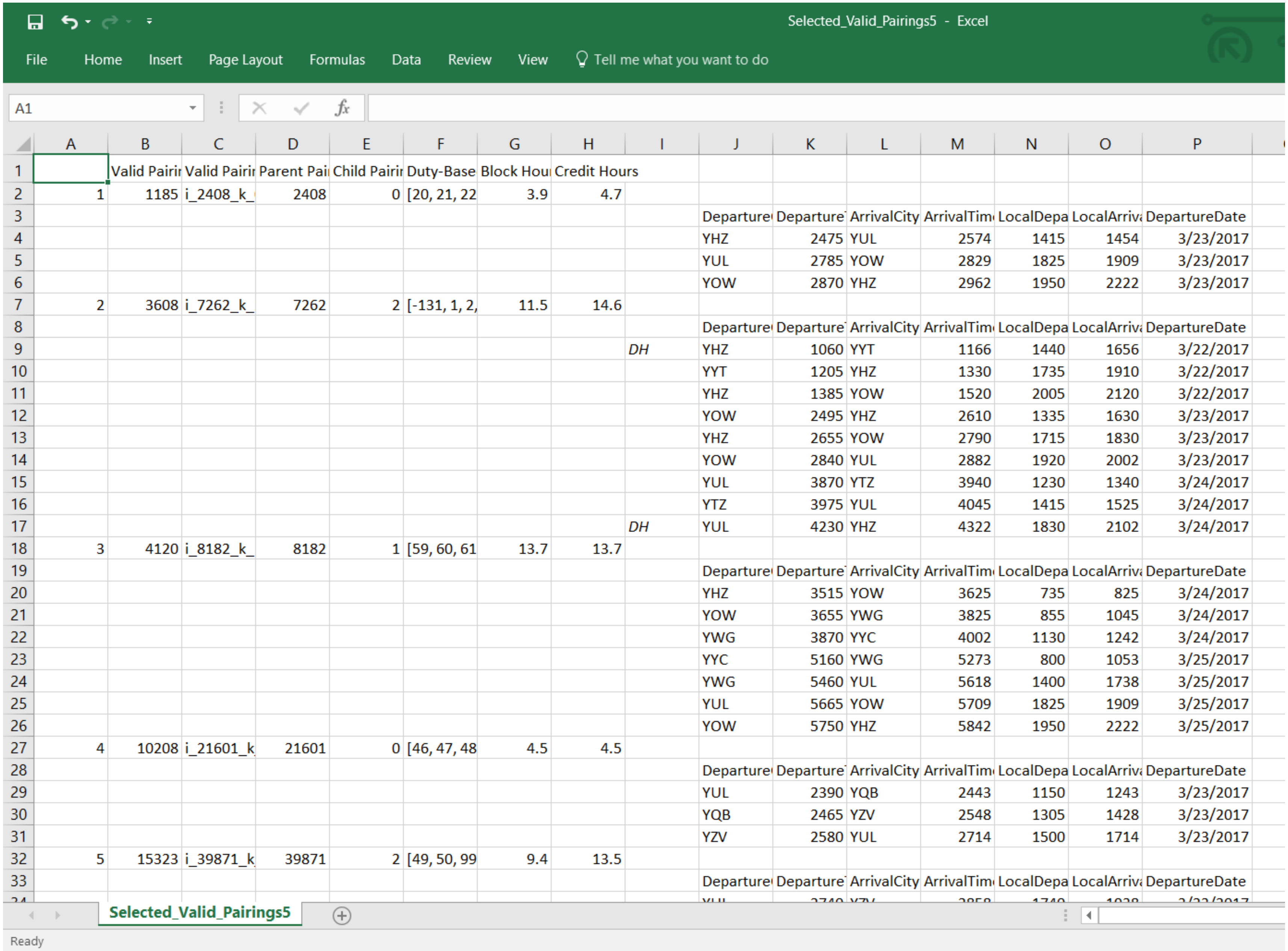
s.t.

$$\sum_{j=1}^{n'} A_{ij} \cdot x_j = 1, \qquad \forall i \in \{1, \ldots, m\} \tag{3.19}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in \{1, \ldots, n'\} \tag{3.20}$$

The objective function (Equation 3.18) minimizes total credit loss while giving preference to shorter deadhead segments as well as lower ratio of total number of deadhead segments used across all of the selected pairings to number of open legs (i.e., lower number of deadhead segments in the solution). Constraint 3.19 ensures that each open leg is covered exactly once.

This marks the end of Phase 2, and thus, the two-phase approach to open-time pairing optimization for cabin crew is completed at this point. Chapter 4 is dedicated to experiments, results and relevant discussions.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Valid Pairi	Valid Pairi	Parent Pai	Child Pairi	Duty-Base	Block Hou	Credit Hours								
2	1	1185	i_2408_k_	2408	0	[20, 21, 22	3.9	4.7								
3										Departure	Departure	ArrivalCity	ArrivalTim	LocalDepa	LocalArriv	DepartureDate
4										YHZ	2475	YUL	2574	1415	1454	3/23/2017
5										YUL	2785	YOW	2829	1825	1909	3/23/2017
6										YOW	2870	YHZ	2962	1950	2222	3/23/2017
7	2	3608	i_7262_k_	7262	2	[-131, 1, 2,	11.5	14.6								
8										Departure	Departure	ArrivalCity	ArrivalTim	LocalDepa	LocalArriv	DepartureDate
9									*DH*	YHZ	1060	YYT	1166	1440	1656	3/22/2017
10										YYT	1205	YHZ	1330	1735	1910	3/22/2017
11										YHZ	1385	YOW	1520	2005	2120	3/22/2017
12										YOW	2495	YHZ	2610	1335	1630	3/23/2017
13										YHZ	2655	YOW	2790	1715	1830	3/23/2017
14										YOW	2840	YUL	2882	1920	2002	3/23/2017
15										YUL	3870	YTZ	3940	1230	1340	3/24/2017
16										YTZ	3975	YUL	4045	1415	1525	3/24/2017
17									*DH*	YUL	4230	YHZ	4322	1830	2102	3/24/2017
18	3	4120	i_8182_k_	8182	1	[59, 60, 61	13.7	13.7								
19										Departure	Departure	ArrivalCity	ArrivalTim	LocalDepa	LocalArriv	DepartureDate
20										YHZ	3515	YOW	3625	735	825	3/24/2017
21										YOW	3655	YWG	3825	855	1045	3/24/2017
22										YWG	3870	YYC	4002	1130	1242	3/24/2017
23										YYC	5160	YWG	5273	800	1053	3/25/2017
24										YWG	5460	YUL	5618	1400	1738	3/25/2017
25										YUL	5665	YOW	5709	1825	1909	3/25/2017
26										YOW	5750	YHZ	5842	1950	2222	3/25/2017
27	4	10208	i_21601_k	21601	0	[46, 47, 48	4.5	4.5								
28										Departure	Departure	ArrivalCity	ArrivalTim	LocalDepa	LocalArriv	DepartureDate
29										YUL	2390	YQB	2443	1150	1243	3/23/2017
30										YQB	2465	YZV	2548	1305	1428	3/23/2017
31										YZV	2580	YUL	2714	1500	1714	3/23/2017
32	5	15323	i_39871_k	39871	2	[49, 50, 99	9.4	13.5								
33										Departure	Departure	ArrivalCity	ArrivalTim	LocalDepa	LocalArriv	DepartureDate

Figure 3.2: Screenshot of the developed decision tool

Chapter 4

Experiments, Results and Discussions

The developed approach and application have been verified using the following data provided by the airline for several 4-day time horizons:

1. open segments;
2. open pairings across the different crew bases;
3. legs making up each of the previously published pairings, some of which have gone into open-time; and
4. deadhead flights available through different airlines from/to each crew base.

The decision support system for crew pairings has been developed in a way that facilitates making modifications when necessary or running experiments. This feature can especially be advantageous when updates to the collective agreement affect the open-time pairing problem at the airline for reserve cabin crew.

4.1 Multiple Pairing Types

Results of several test runs are summarized in Tables 4.1 and 4.2. Table 4.1 shows the various types of pairings achieved by the developed crew pairing application. It should be noted that when verifying results with multiple duty and pairing types, the type of duty/pairing must be taken into consideration because it influences the constraints to be satisfied.

4.2 Duration of Layovers and Sit Connections

With regards to verifying that none of the constraints given in Chapter 3 have been violated within the pairing generator/optimizer solution, Table 4.2 addresses sample constraints. It can be seen that in the optimal solution regarding each dataset,

Dataset	Number of open legs	Total number of selected feasible pairings	Number of Type 1-A and Type 1-B pairings	Number of Type 1-C pairings	Number of Type 2 pairings
Dataset 1	20	7	0	7	0
Dataset 2	27	7	3	2	2
Dataset 3	38	9	2	3	4
Dataset 4	38	5	0	0	5
Dataset 5	39	11	3	6	2
Dataset 6	46	14	1	12	1
Dataset 7	65	12	0	6	6
Dataset 8	68	16	3	7	6
Dataset 9	75	20	2	12	6
Dataset 10	80	22	3	14	5
Dataset 11	87	18	4	6	8

Table 4.1: Pairing types within the optimal selection of pairings for several sample runs

the constraints governing the number of duties (Equation 4.1) and the length of rest periods (Equation 4.2) in each crew pairing are respected. In what follows, limitations on the duration of rest periods and sit connections will be examined more closely.

$$\text{number of duties in pairing} \leq 4\,(= \text{MAX-NUMBER-OF-DUTIES}) \qquad (4.1)$$

$$\text{length of layover in pairing} \geq 10 \text{ hrs } (= \text{MIN-LENGTH-OF-LAYOVER}) \qquad (4.2)$$

4.2.1 Adding a New Constraint: Setting an Upper Bound on Sit Connections

In this case study, the collective bargaining agreement (CBA) between the union of flight attendants and the airline does not enforce an upper limit for the duration of rest periods between duties as well as the duration of sit connections between flight

Dataset	Number of open legs	Number of selected multi-day pairings	Maximum number of duties in pairings	Minimum length of layover in pairings	Maximum length of layover in pairings
Dataset 1	20	0	1 duty	0	0
Dataset 2	27	2 pairings	4 duties	14.3 hrs	24.4 hrs
Dataset 3	38	9 pairings	2 duties	11.4 hrs	19.3 hrs
Dataset 4	38	5 pairings	3 duties	11.4 hrs	20.3 hrs
Dataset 5	39	2 pairings	3 duties	15.3 hrs	17.6 hrs
Dataset 6	46	1 pairing	2 duties	14.7 hrs	14.7 hrs
Dataset 7	65	6 pairings	3 duties	13.5 hrs	19.3 hrs
Dataset 8	68	6 pairings	3 duties	11.0 hrs	17.7 hrs
Dataset 9	75	6 pairings	3 duties	11.5 hrs	20.3 hrs
Dataset 10	80	5 pairings	3 duties	10.1 hrs	19.5 hrs
Dataset 11	87	8 pairings	3 duties	12.2 hrs	18.6 hrs

Table 4.2: Rest periods or layovers between duty periods for the optimal selection of pairings

segments. The effect of adding such a constraint to the crew pairing problem, particularly regarding sit connections, will be examined through several experiments on one dataset. Among the datasets at hand, Dataset 1 is used for this purpose, because, as can be seen in Table 4.1, the solution is conveniently composed of single-day Type 1-C pairings. Therefore, all of the pairings in the solution to the original unbounded problem are governed by the same constraints and thus, similar rules with regards to sit connections. However, some of the experiments are expected to change the structure of the original solution while others are not. The experiments are carried out in the following manner.

Firstly, once the optimal solution of the original *unbounded problem* is obtained, the longest sit connection appearing within the entirety of duty periods is found. In what follows, this value will be called *Longest-Sit-Found*. The second longest sit connection is also used in this experiment and will be referred to as *2nd-Longest-Sit-Found*. Note that the *unbounded problem* refers to the pairing generation problem that does not constrain the maximum permitted length of sit connections between consecutive legs (Equation 4.3).

$$\text{length of sit connection} \leq \text{MAX-LENGTH-OF-SIT} \tag{4.3}$$

Having determined the aforementioned values, the constraint given in Equation 4.3 is added to the pairing generation module. Next, in each experiment, parameter MAX-LENGTH-OF-SIT is set to a different value determined by Longest-Sit-Found and 2nd-Longest-Sit-Found as follows:

1. MAX-LENGTH-OF-SIT $\gg$ Longest-Sit-Found
2. MAX-LENGTH-OF-SIT $>$ Longest-Sit-Found
3. MAX-LENGTH-OF-SIT $=$ Longest-Sit-Found
4. 2nd-Longest-Sit-Found $<$ MAX-LENGTH-OF-SIT $<$ Longest-Sit-Found
5. MAX-LENGTH-OF-SIT $=$ 2nd-Longest-Sit-Found
6. MAX-LENGTH-OF-SIT $<$ 2nd-Longest-Sit-Found

Eventually, for each case, the pairing generator and optimizer are run again and solutions are compared with that of the original unbounded problem. The described experiments are run on Dataset 1, where the longest and second longest sit connections found as well as total credit loss of achieved pairings are as follows:

Longest-Sit-Found = 196 minutes (or 3.3 hours)
2nd-Longest-Sit-Found = 129 minutes (or 2.2 hours)
Total Credit Loss of Achieved Pairings = 0.026 (or 2.60%)

Results of the aforementioned experiments are summarized in Table 4.3. It is worth mentioning that in these experiments, a significant change in run time was not observed. Table 4.3 shows that as expected, reduction in the feasible region results in the exclusion of pairings that were once legal in the original unbounded problem. Thus, the minimization of the total credit loss gets worse. These results emphasize the fact that in an optimization problem with a bounded feasible region, constraints leading to an even more limited set of feasible solutions are to be incorporated only

Experiment	Description	MAX-LENGTH-OF-SIT (minutes)	Total Credit Loss of Achieved Pairings	Overall Assessment of Solution Compared with Unbounded Problem
1	$\gg$ 196 minutes	250	0.026	no change
2	$>$ 196 minutes	197	0.026	no change
3	$=$ 196 minutes	196	0.026	no change
4	$>$ 129 and $<$ 196 minutes	195	0.133	slightly worse
5	$=$ 129 minutes	129	0.133	slightly worse
6	$<$ 129 minutes	128	1.369	significantly worse

Table 4.3: Experiments on a sample dataset regarding the duration of sit connections

when required. For this reason, in the developed pairing optimization tool, the constraint given in Equation 4.3, which can be seen in airline crew pairing literature, is not applied, even with a very large value of parameter MAX-LENGTH-OF-SIT. However, in the event that the airline decides to select an upper limit for the duration of intermissions between flight segments and/or duties, the underlying idea behind the experiments summarized in Table 4.3, i.e., keeping track of the maximum sit connection and/or layover observed in the optimal solution to the pairing problem over time, will be beneficial.

4.3 Varying Problem Size

Table 4.4 provides insight as to how big a typical open-time crew pairing problem for reserve flight attendants is with respect to all 5 crew bases of the airline under study. Historical records show that the total number of open legs from all sources, namely open segment report and open pairing report, is normally below 100 legs over the course of 4 days. Thus, the problems typically encountered by the "Next-day Crew Scheduling Department" at the airline can be solved in a reasonable amount of time on mainstream computers using current LP/IP solvers, which is far beyond the speed of manual scheduling. Application run time is especially important in such a problem that targets the recovery of crew schedules at the operational level. Before

Datasets	Number of open legs	Total number of deadheads available	Number of constructed *ppairings*	Number of valid pairings
Dataset 1	20	813	4798	2841
Dataset 2	27	5903	25814	7603
Dataset 3	38	6020	71183	37524
Dataset 4	38	6132	224642	150017
Dataset 5	39	6400	51563	26861
Dataset 6	46	6312	27981	10607
Dataset 7	65	6077	168001	73818
Dataset 8	68	2866	137447	44291
Dataset 9	75	6002	548970	197932
Dataset 10	80	4915	267966	62183
Dataset 11	87	2591	514904	128636

Table 4.4: Sample Runs: Relationship between the number of given open legs, number of available deadhead legs from/to bases, number of constructed *potential pairings*, and number of valid *duty-based pairings* across all crew bases

presenting sample run times, the overall impact of *potential pairing* construction at the beginning of Phase 1 on total program execution time will be illustrated.

The value of the preprocessing procedure described in Section 3.1.1 can be understood by evaluating how much the problem size is reduced early on in the process of generating all possible valid pairings given a specific number of open segments. Consider a problem involving 68 open legs over the 4-day planning horizon. From a set theory perspective, the crew pairing construction problem is, in a way, similar to finding all of the possible subsets, which respect work rules, for the set of open legs. Clearly, in this context, the subset containing zero elements, i.e., no legs, is of no interest. Hypothetically, if there were absolutely no constraints as to how flight segments could be grouped together to form pairings, combinatorics suggests that there would be at most $2^{68} - 1 = 295,147,905,179,352,825,855$ possible pairings, the lengths of which, in terms of number of legs, would range from 1 to 68. Undoubtedly, the notion behind the logical constraints given in Table 3.1 alone suggests that many of these subsets are invalid. This is due to the fact that if, either time-wise or location-wise, a sequence of two arbitrary segments is infeasible, then this sequence, as is, must not appear in any *potential pairing*. As it can be seen in Table 4.4, for a sample problem with 68

Datasets	Number of open legs	Total number of deadheads available	Valid pairing generator run time (sec)	Pairing optimizer run time (sec)
Dataset 1	20	813	1.2	0.5
Dataset 2	27	5903	11.0	0.9
Dataset 3	38	6020	39.9	3.7
Dataset 4	38	6132	192.9	15.8
Dataset 5	39	6400	36.7	2.9
Dataset 6	46	6312	22.4	1.4
Dataset 7	65	6077	91.5	11.4
Dataset 8	68	2866	59.5	7.5
Dataset 9	75	6002	210.4	35.7
Dataset 10	80	4915	166.8	11.3
Dataset 11	87	2591	114.2	23.0

Table 4.5: Sample run times (in seconds)

open legs, the number of *potential pairings*, or *ppairings*, is incomparable with the number of unconstrained possibilities suggested by Combinatorial Mathematics on the first look. Thus, incorporating such a preprocessing module within the developed application conveniently leads to less operations and run time required to generate all feasible pairings which build the foundation for finding the optimal selection of pairings.

Table 4.5 shows sample run times for valid pairing generation as well as pairing optimization. The feasible pairing generator has been programmed in the Python programming language using Python's Integrated DeveLopment Environment (IDLE) Version 2.7. After this module is run, relevant data is exported to the optimization phase in matrix form through CSV, or Comma Separated Values, files. Figure 4.1 illustrates the integration of these modules and files. The pairing optimizer uses open-source GUSEK Version 0.2 to model the pairing selection problem in GNU MathProg Language (GMPL), which is solved using GLPK solver. This model can be found in Appendix C. Sample runs have been performed on an Intel® Core™ i5-6300HQ CPU @2.30 GHz processor/8 GB RAM machine. As it can be seen in Table 4.5, depending on the dataset and problem size, total execution time typically ranges between a few seconds to a few minutes.

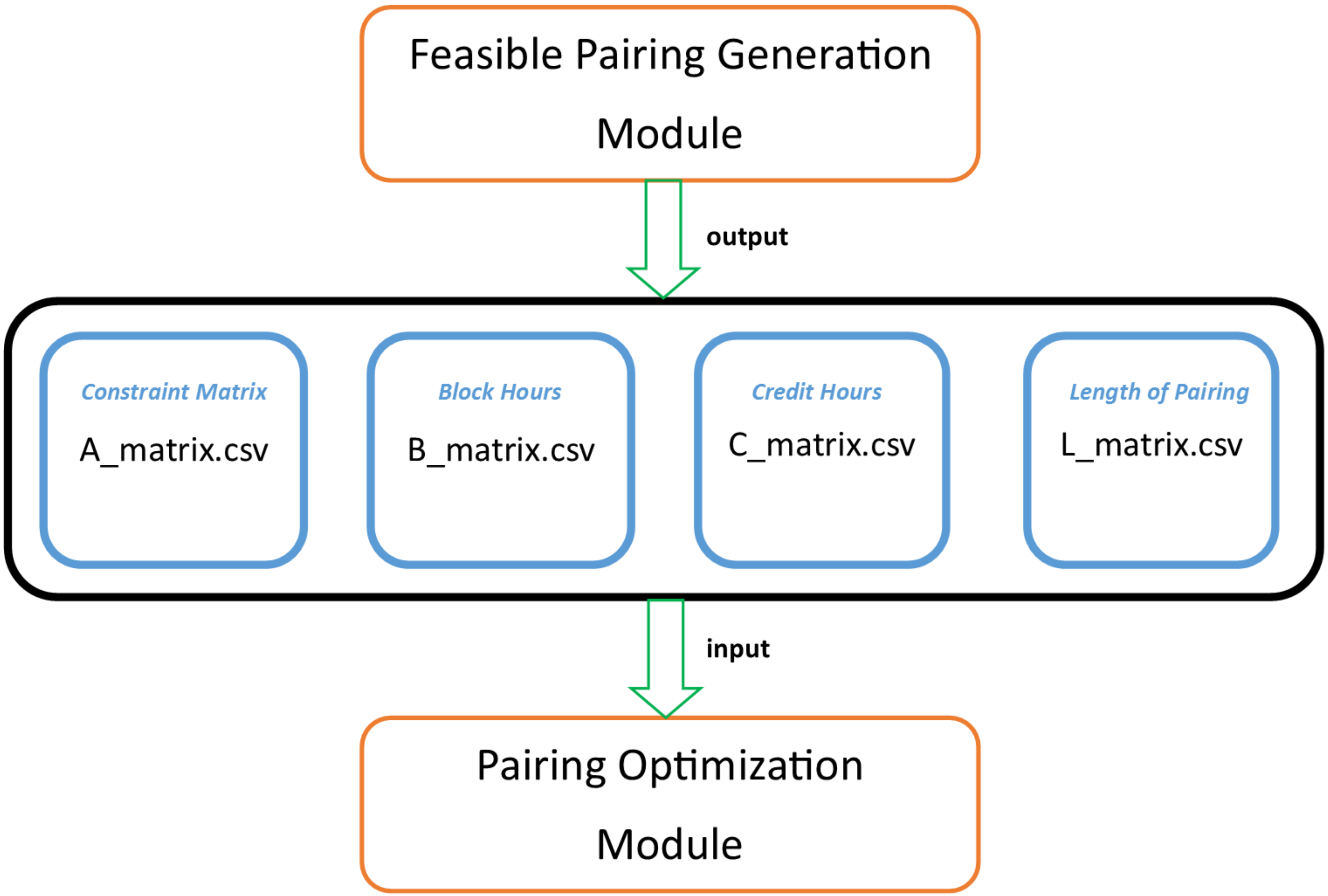


Figure 4.1: Interaction between Feasible Pairing Generator and Pairing Optimizer

It has been observed that GUSEK is not capable of handling relatively large open-time pairing optimization problems, which are less likely to be encountered at the airline but may still be seen occasionally. To be specific, attempts to solve the optimization model for the datasets described in Table 4.6 using GLPK solver resulted in memory error. To overcome this issue, the pairing optimization model has been translated to MPL Modeling Language and solved using Gurobi Solver. It is worth noting that in case of Dataset B, the program was run on a machine with a RAM size of 32 GB; however, less than half of the amount of available memory was used. It can be concluded that GLPK/GUSEK performs well for regular-sized problems (roughly under 100 open legs), i.e., the optimal solution is reached in a reasonable amount of time. However, larger problems (roughly over 100 open legs) may need to be solved using a more powerful LP/IP solver such as Gurobi.

Large Dataset	Number of open legs	Total number of deadheads available	Total number of constructed *ppairings*	Total number of valid pairings	Pairing optimizer run time
A	106	6,296	1,127,200	208,830	1.32 sec
B	106	6,247	$>$ 5 million	$>$ 1 million	12.81 sec

Table 4.6: Optimization run times of large datasets

4.4 Optimal Credit Loss

Optimization results showing total block hours and total credit hours for selected legal pairings in several test runs are given in Table 4.7. Credit loss, minimized by solving the binary integer programming model of the second phase to optimality, shows how far actual flying times are from achieved credit hours based on which flight attendants are scheduled to get paid. Table 4.8 depicts each individual optimal pairing.

The objective of the optimization problem has been defined as the minimization of total credit loss across all pairings, suggesting that theoretically, if necessary, inefficient pairings may appear in the optimal solution as long as the negative impact of such pairings on the overall difference between credit hours and block hours is

	Datasets										
	1	2	3	4	5	6	7	8	9	10	11
Number of open legs	20	27	38	38	39	46	65	68	75	80	87
Total Block Hours (hrs)	38.4	40.6	50.6	61.2	69.9	57.2	89.3	84.5	118.3	103.6	121.2
Total Credit Hours (hrs)	39.4	53.9	67.6	68.9	55	74.4	106.6	123	144.9	145	149.6
$\frac{\sum CreditHours}{\sum BlockHours}$	1.025	1.328	1.336	1.126	1.271	1.300	1.194	1.456	1.225	1.400	1.234
Credit loss	0.025	0.328	0.336	0.126	0.271	0.300	0.194	0.456	0.225	0.400	0.234

Table 4.7: Optimal results

compensated for by more efficient pairings. Taking a closer look at each individual pairing in the solutions of sample runs verifies that regardless of the percentage of total credit loss, it is not beyond expectation to see pairings with absolutely no credit loss. In fact, Table 4.8 shows that it is possible for a fairly high percentage of the pairings to have a credit loss of $1^{+5\%}_{-0}$.

Dataset	Number of open legs	Number of selected pairings	Average pairing $\frac{CH}{BH}$	Minimum pairing $\frac{CH}{BH}$ achieved	Number of pairings with minimum ratio ($1.00^{+0.05}_{-0}$)
Dataset 1	20	7	1.03	1.00	6
Dataset 2	27	7	2.13	1.06	3
Dataset 3	38	9	1.64	1.00	3
Dataset 4	38	5	1.29	1.00	2
Dataset 5	39	11	1.78	1.00	5
Dataset 6	46	14	1.43	1.00	4
Dataset 7	65	12	1.22	1.00	5
Dataset 8	68	16	1.58	1.00	1
Dataset 9	75	20	1.35	1.00	10
Dataset 10	80	22	2.01	1.00	7
Dataset 11	87	18	1.38	1.00	8

Table 4.8: A glance at the ratio of credit hours to block hours ($\frac{CH}{BH}$) for individual selected pairings in obtained solutions to optimization problem across all crew bases

4.5 Effects of Deadheads on Pairings

It must be noted that deadheads do not show up in block hour calculations and only contribute to pairing credit hours. Therefore, deadheads are considered costly for the airline even though the credit hours awarded on a deadhead segment is only a percentage of the credit hours of an actual productive flight. One reason why airlines prefer less deadheads is the limited number of legs permitted within each duty period/pairing, and deadheads count as legs. However, deadheads play an important role in creating feasible pairings. In other words, deadheading is generally inevitable in open-time pairing construction, mainly due to the constraint that requires all legal pairings to begin and terminate at a crew base. For this purpose, a lengthy list of flights on which, if necessary, cabin crew may fly as passengers is required for identifying useable deadheads. Therefore, depending on what flights are available for this purpose during the planning period, segments that are appropriate in terms of origin/destination as well as departure time/arrival time are identified. The following example illustrates this concept.

Consider a pairing where an anonymous crew member would begin the pairing by flying on a deadhead segment as a passenger from their home base to another airport where they would start working on a sequence of legs and eventually fly back home. To ensure that they would arrive at their first working airport in time in the event of unexpected delays, a minimum buffer time, for instance 30 minutes, is added prior to the departure of the first actual leg in the pairing. Therefore, the following relationships must hold true with regards to connecting a deadhead segment and an actual flight on the basis of time (Equation 4.4) and location (Equation 4.5):

$$ArrivalTime[DeadheadFromBase] + \text{BUFFER-TIME} < DepartureTime[Leg_1] \tag{4.4}$$

$$DestinationCity[DeadheadFromBase] = OriginCity[Leg_1] \tag{4.5}$$

4.5.1 Feasible Region of Optimization Problem and Dummy Pairings

Wherever necessary, the pairing generator refers to the list of useable deadheads in order to make it possible for *potential pairings* to begin at a base and end at the same base. Since the final goal is to find a subset of feasible pairings that cover all open legs at minimum cost, in order for the optimal solution to exist, it must be ensured that the feasible region is not empty in the first place, i.e., it is necessary that each open leg appear in at least one valid pairing.

Theoretically, for the simplified problem consisting of one crew base, a basic solution to the Set Partitioning Problem formulation of Phase 2, provided in Chapter 3, is the solution that is composed of single-leg pairings which are surrounded by deadheads from or/and back to base as needed. It can easily be shown that such pairings, which either consist of 2 or 3 legs, are legal. In the worst case, the described solution which satisfies the leg covering constraint and thus is feasible, is also optimal provided that no better subset of legal pairings exists. However, practically speaking, there may be instances where due to the lack of desirable deadhead flights from/to crew bases, certain open legs do not show up in any feasible pairing, which eventually leads to such segments remaining in open-time. Before explaining how the optimization model of the developed application deals with this issue and what the airline can do to resolve it, it is worth noting that such instances are not unlikely to happen; however, given the list of open segments along with flights available for crew deadheading, the algorithm given in Appendix E can be used to predict exactly which legs will be left uncovered.

The developed decision support system makes sure that at least one feasible solution exists for the pairing optimization problem by adding dummy pairings with very high costs. At the end of Phase 1, once all valid pairings are found, for every open segment, a dummy pairing solely consisting of that open leg at a very large cost, i.e., infinity block hours and credit hours and pairing length in theory, is created and appended to the list of valid pairings passed on to Phase 2. When the minimization model searches for a subset of pairings to satisfy the leg covering constraint, it will not select from the described single-leg dummy pairings unless it is absolutely necessary.

4.5.2 Legs Remaining in Open-time

The size and quality of the list of deadheading options directly impact the optimization solution. With regards to the optimal solution to the pairing selection problem, the post-processing module introduced earlier distinguishes between dummy pairings, that show which legs remain in open-time, and selected feasible pairings, based on which total block hours and total credit hours are compared.

It must be noted that among the described open legs, may be segments that depart from the second planning day onwards. Therefore, due to the rolling time horizon of *next-day scheduling* at the airline, in such cases, it is probable that solving for the following 4-day period consisting of a new pool of open flights will fit some of these legs in feasible pairings. In case an uncovered leg departs on the first day of the planning period, i.e., the day after schedules are published, the airline's *day-of crew scheduling* team that are basically in charge of finding solutions to schedule disruptions occurring on the day of operations, take over.

4.6 Defining Metrics to Evaluate the Solution

Using the tables to follow, sample solutions from three test runs will be evaluated in terms of various metrics to assess individual pairings as well as the entire set of selected feasible pairings. Table 4.9 summarizes the use of deadheads in the optimal solutions obtained by running the pairing optimizing application for several datasets. Note that whether or not deadheading is part of a solution highly depends on the nature of the open segments given as input to the crew pairing problem. However, what is certain is that for a given scheduling period, the total number of deadhead flights used in the solution, which make commencing and terminating at crew bases possible, fall between 0 (best case) and 2 × *number of open legs* (worst case). For example, in case of Dataset 11, with 87 open legs in the input, 7 deadhead segments have appeared in the solution, which means on average there are $\frac{7}{87}$ or 0.08 deadhead segments to every open segment (i.e., 1 deadhead for every 12 open legs). Airlines naturally prefer solutions with low ratio of deadhead segments to number of open legs denoted by ρ and given in Equation 4.6. Despite the fact that a solution may not be very appealing in every way, it should be noted that the described solution is the solution that leads to the lowest total cost, i.e., credit loss, while respecting rules and regulations regarding cabin crew and is thus considered acceptable by the airline. Nonetheless, the metrics defined to describe the optimal solution simply illustrate the tradeoff between lower cost and other factors of interest.

It must be recognized that from the airline crew planning perspective which seeks relatively desirable solutions on top of low-cost solutions, certain factors are especially important to assess over time. Therefore, once the minimum-cost set of pairings is obtained, evaluating the solution based on defined metrics provides the means for quantifying *solution desirability*. In what follows, another metric defined with regards to crew deadheading (μ given in Table 4.9) will be explained.

Dataset	Number of open legs	Number of available dead-heads	Total number of selected feasible pairings	Number of feasible pairings with dead-heading	Total number of deadhead flights used in solution	ρ	μ
1	20	813	7	0	0	0	0 %
2	27	5,903	7	1	1	0.04	14 %
3	38	6,020	9	3	4	0.11	33 %
4	38	6,132	5	1	1	0.03	20 %
5	39	6,400	11	3	6	0.15	27 %
6	46	6,312	14	3	3	0.07	21 %
7	65	6,077	12	2	3	0.05	17 %
8	68	2,866	16	4	6	0.09	25 %
9	75	6,002	20	4	5	0.07	20 %
10	80	4,915	22	9	11	0.14	41 %
11	87	2,591	18	5	7	0.08	28 %

Table 4.9: Sample results showing the count of deadheads in the obtained solutions

$$0 < \rho = \frac{\text{total number of deadhead segments in solution}}{\text{number of open legs}} < 2 \tag{4.6}$$

As Table 4.9 suggests, among the several feasible pairings selected by the optimizer, there are pairings with deadheading at the beginning and/or end of the sequence of legs they cover, which eventually need to be assigned to flight attendants through crew rostering. This piece of information may especially be important for crew members bidding on published pairings as some flight attendants may or may not prefer deadheading for personal reasons. From the planner's perspective, the percentage of pairings that include deadheading (DH), denoted by μ, can be calculated (Equation 4.7) and compared against specified target values in order to assess the attractiveness of the obtained solution.

$$\mu = \% \text{ of pairings with DH} = \frac{\text{number of selected pairings with DH}}{\text{total number of selected pairings}} \times 100 \tag{4.7}$$

Chapter 5

Conclusions

Existing research have mostly examined the crew scheduling problem at the strategic level, specifically focusing on modeling and finding near-optimal or optimal solutions to the crew pairing problem that is required to be solved at airlines as part of the typical pre-month planning process. On the other end of the planning spectrum, lies the crew rescheduling problem at the operational level, which aims to recover crew schedules from disruptions that may occur during daily operations. Compared to the monthly problem, the crew rescheduling problem is a less-studied area, although after initial schedules are published, changes in crew availability, flight schedules, and aircraft are inevitable, resulting in what is known as open-time flying, and thus, a proper approach to this time-critical problem can prevent unnecessary extra costs for airlines.

Previous research in this area have mainly addressed the rescheduling problem encountered on the day of operations. However, open-time flying may be caused by various sources, some of which make it possible to identify uncovered legs and/or pairings several days prior to the day of operations, therefore allowing for a significant portion of disruptions to be dealt with in a less time-restrained planning environment by utilizing reserve crews as efficiently as possible. This aspect of crew rescheduling is overlooked by previous academic research. This study, done in collaboration with industry partner, Jazz Aviation LP, aimed at designing a decision support system to help the Scheduling Department re-optimize the generation of legal open-time pairings for reserve cabin crew, and has led to the proposal of a two-phase approach to this problem.

Since crew unavailability itself is a source of open-time flying, the crew rescheduling problem on the next-day level is advised to be carried out in two separate stages in a similar fashion as the monthly problem, namely crew pairing and crew assignment. As airline requirements vary substantially, existing modeling techniques for the crew pairing problem could not be adapted directly. The key characteristics of our pairing problem, where optimal legal pairings are generated independent of individual reserve crew availability or preferences, include multiple duty types, multi-day rolling time horizon, complex non-linear crew pay structure in terms of credit hours, multiple crew bases, and use of actual flights available for deadheading crew from and/or to base.

Given a list of flight segments in open time as well as flights that may be used in crew deadheading where needed, following data processing and cleaning, our proposed enumeration-based technique begins with the construction of valid pairings in Phase 1. The underlying assumption is that all flight attendants are qualified to work on all aircraft types, thus fleet types have no impact on the pairing generation problem and are ignored. The different restrictions that are encountered when building pairings that can feasibly be flown by cabin crews fall into two major groups, (a) logical constraints, and (b) airline-specific constraints.

Phase 1 begins with a preprocessing procedure that addresses logical constraints by building what we have called *potential pairings*. Incorporating a preprocessing module within the developed application conveniently leads to less operations and run time required to generate all valid pairings. In order to apply the constraints to the constructed *potential pairings*, due to the fact that certain parameters in the collective agreement depend on duty type, it is first necessary to identify duty periods within each *potential pairing*. The duty period identification process is carried out by examining the *Leg-Gap* pattern associated with each *potential pairing* in order to produce *duty-based pairings*, which are the sequences that the airline-specific constraints can directly be applied to once the type of pairing and its duty periods are determined. Thus, for every resulting *duty-based pairing*, the type of each pairing as well as duty period if applicable (i.e., in case of multi-day as opposed to single-day pairings), is specified. The validation procedure then filters out invalid *potential pairings*. The costs of all valid *duty-based pairings* in terms of block hours and credit hours are calculated at the end of Phase 1, which allow for the removal of non-linearity from the optimization model of Phase 2.

Valid pairings and their costs are passed to Phase 2 in order to select a set of pairings that minimizes total credit loss while covering each open segment once. The optimization problem is formulated as a binary integer programming (BIP) formulation, namely Set Partitioning Problem (SPP) formulation, with the exception that the objective function is modified to incorporate deadheading preferences. The length of pairings is used to select the most desirable pairing among pairings with similar credit loss.

The proposed two-phase approach has been verified using data provided by the airline. Depending on the dataset and problem size, total execution time typically ranges between a few seconds to a few minutes. We conclude that GLPK/GUSEK performs well for regular-sized problems ($\sim$ under 100 open legs), i.e., the optimal solution is reached in a reasonable amount of time. Larger problems (100+ open legs) may need to be solved using more powerful LP/IP solvers such as Gurobi. While we have tackled the gap existing in research between the monthly problem and the day of operations airline crew scheduling problem, we have been able to build a decision support system to assist crew schedulers in practice, which has been successful in finding optimal solutions to the open-time crew pairing problem for reserve flight attendants in a reasonable amount of time. This is much faster than a purely manual approach. In addition, we find the optimal selection of pairings by considering all of the valid pairings for all crew bases. Considering all crew bases simultaneously results in a better objective function (i.e., lower credit loss) compared to a sequential approach that looks at the pairing problem separately for each base. Our system also considers a lengthy list of available deadhead segments that is impossible to efficiently work with manually.

This research project is recommended to be extended in the following ways:

- *Adding deadhead legs in the middle of pairings:*

We have used deadheading only at the beginning and end of pairings. However, in practice, it is possible to deadhead crews in the middle of pairings if this leads to lower total credit loss. Therefore, one way to extend this work is to consider deadhead flights in the middle of pairings in addition to the beginning and end.

- *Adding other deadheading options used in practice (e.g., ground transportation):* This project has been carried out based on the assumption that the only way to transport crews from base to another city or back to base is by flying them on flights available for deadheading. Under this assumption, how crew deadheading via air affects pairing block hours and credit hours were known. In practice, other options for the purpose of transporting crews as passengers, such as ground transportation, are common. Once the effect of such transportation modes on constraints regarding duty and pairing construction as well as associated block hours and credit hours is known with regards to crew deadheading, additional deadheading options may be incorporated in the pairings.

- *Incorporating an estimate of available reserve crews at each base (number, availability, etc.) within Phase 2:*
  The ultimate solution to the crew pairing problem is an optimal set of pairings to be used in crew assignment. In our pairing problem, the solution is optimal with respect to all of the crew bases at the airline without considering reserve crew availability at each base. In order to come up with a more practical solution, information regarding crews that are on call during the planning period, such as an estimate of the number of available reserve crews at each base as well as the number of days each crew is available, is recommended to be incorporated within the optimization model of Phase 2.

- *Updating minimum sit connection time between legs in each duty period when aircraft tail numbers data becomes available:*
  In order for a cabin crew to connect from one leg to the following segment in their duty, a minimum sit connection time is required. In practice, if the crew does not need to change aircraft, then the required amount of sit connection is less than cases where they would need to switch aircraft. Considering this may

lead to better pairings and thus improve total credit loss; however, an integrated data report containing aircraft tail numbers must be available for this purpose. When this information becomes accessible in a useable format, we recommend updating the minimum required sit connection used in the construction of valid open-time pairings at the airline.

- *Improving the pre-processing procedure to reduce the number of potential pairings to be validated in Phase 1:*

  At the beginning of Phase 1, we generate *potential pairings*, which are sequences of legs that respect the logical location and time constraints. The time constraint we have specifically applied states that two segments may follow one another in a pairing (and/or duty period) only if the first leg arrives before the departure of the next leg while allowing for sufficient time for crew to transition between flights, which is MIN-SIT. Therefore, in an attempt to construct every sequence that would logically be acceptable, even at a clearly high cost, we have not imposed an upper bound on sit connection times. However, we observe that with regards to each crew base, numerous distinct *potential pairings* frequently cover the exact same open leg(s) and only differ in deadhead segments that belong to different times/dates, leading to unnecessary and expensive *potential pairings*. (This is because our planning period spans multiple days and we have a large pool of deadhead flights at hand across the airline's multiple crew bases.)

  We suggest improving the pre-processing procedure to identify and eliminate the *potential pairings* that would clearly be found infeasible eventually and/or the ones that, if found feasible and passed to the optimization phase, would not be able to compete with other *potential pairings* under any circumstances. This extension will reduce the amount of memory used by the valid pairing generation code of Phase 1, which is particularly important for large datasets.

Bibliography

[1] About Jazz. http://www.flyjazz.ca/en/home/aboutjazz/default.aspx. Retrieved: 2017-06-22.

[2] Jazz Aviation Corporate Fact Sheet. http://www.flyjazz.ca/en/home/aboutjazz/corporatefactsheet/default.aspx. Retrieved: 2017-06-22.

[3] Shervin AhmadBeygi, Amy Cohn, and Marshall Weir. An integer programming approach to generating airline crew pairings. *Computers & Operations Research*, 36(4):1284–1298, 2009.

[4] Ranga Anbil, Eric Gelman, Bruce Patty, and Rajan Tanga. Recent advances in crew-pairing optimization at american airlines. *Interfaces*, 21(1):62–74, 1991.

[5] Erik Andersson, Efthymios Housos, Niklas Kohl, and Dag Wedelin. Crew pairing optimization. *Operations research in the airline industry*, pages 228–258, 1998.

[6] JP Arabeyre, J Fearnley, FC Steiger, and W Teather. The airline crew scheduling problem: A survey. *Transportation Science*, 3(2):140–163, 1969.

[7] Edward K Baker, Lawrence D Bodin, William F Finnegan, and Ronny J Ponder. Efficient heuristic solutions to an airline crew scheduling problem. *AIIE Transactions*, 11(2):79–85, 1979.

[8] Egon Balas. A class of location, distribution and scheduling problems: Modeling and solution methods. 1982.

[9] Michael Ball and Anito Roberts. A graph partitioning approach to airline crew scheduling. *Transportation Science*, 19(2):107–126, 1985.

[10] Cynthia Barnhart, Levent Hatay, and Ellis L Johnson. Deadhead selection for the long-haul crew pairing problem. *Operations Research*, 43(3):491–499, 1995.

[11] M. Bazargan. *Airline Operations and Scheduling*. Ashgate, 2004.

[12] Paola Cappanera and Giorgio Gallo. A multicommodity flow approach to the crew rostering problem. *Operations Research*, 52(4):583–596, 2004.

[13] Hai D Chu, Eric Gelman, and Ellis L Johnson. Solving large scale crew scheduling problems. *European Journal of Operational Research*, 97(2):260–268, 1997.

[14] Jens Clausen, Allan Larsen, Jesper Larsen, and Natalia J Rezanova. Disruption management in the airline industry –Concepts, models and methods. *Computers & Operations Research*, 37(5):809–821, 2010.

[15] Herbert Dawid, Johannes König, and Christine Strauss. An enhanced rostering model for airline crews. *Computers & Operations Research*, 28(7):671–688, 2001.

[16] Paul R Day and David M Ryan. Flight attendant rostering for short-haul airline operations. *Operations research*, 45(5):649–661, 1997.

[17] Guang-Feng Deng and Woo-Tsong Lin. Ant colony optimization-based algorithm for airline crew scheduling problem. *Expert Systems with Applications*, 38(5):5787–5793, 2011.

[18] Guy Desaulniers, J Desrosiers, Y Dumas, S Marc, B Rioux, Marius M Solomon, and Francios Soumis. Crew pairing at Air France. *European journal of operational research*, 97(2):245–259, 1997.

[19] Viktor Dück, Franz Wesselmann, and Leena Suhl. Implementing a branch and price and cut method for the airline crew pairing optimization problem. *Public Transport*, 3(1):43, 2011.

[20] Walid El Moudani, Carlos Albert Nunes Cosenza, and Félix Mora-Camino. An intelligent approach for solving the airline crew rostering problem. In *Computer Systems and Applications, ACS/IEEE International Conference on. 2001*, pages 73–79. IEEE, 2001.

[21] Walid El Moudani and Félix Mora-Camino. A dynamic fuzzy approach for the evaluation of airlines' crew satisfaction. *International Journal of Combinatorial Optimization Problems and Informatics*, 1(2), 2010.

[22] Torsten Fahle, Ulrich Junker, Stefan E Karisch, Niklas Kohl, Meinolf Sellmann, and Bo Vaaben. Constraint programming based column generation for crew assignment. *Journal of Heuristics*, 8(1):59–81, 2002.

[23] Balaji Gopalakrishnan and Ellis L Johnson. Airline crew scheduling: state-of-the-art. *Annals of Operations Research*, 140(1):305–337, 2005.

[24] Karla L Hoffman and Manfred Padberg. Solving airline crew scheduling problems by branch-and-cut. *Management Science*, 39(6):657–682, 1993.

[25] Niklas Kohl and Stefan E Karisch. Airline crew rostering: Problem types, modeling, and optimization. *Annals of Operations Research*, 127(1):223–257, 2004.

[26] Ladislav Lettovskỳ, Ellis L Johnson, and George L Nemhauser. Airline crew recovery. *Transportation Science*, 34(4):337–348, 2000.

[27] Broos Maenhout and Mario Vanhoucke. A hybrid scatter search heuristic for personalized crew rostering in the airline industry. *European Journal of Operational Research*, 206(1):155–167, 2010.

[28] Claude P Medard and Nidhi Sawhney. Airline crew scheduling from planning to operations. *European Journal of Operational Research*, 183(3):1013–1027, 2007.

[29] Rüdiger Nissen and Knut Haase. Duty-period-based network model for crew rescheduling in European airlines. *Journal of Scheduling*, 9(3):255–278, 2006.

[30] Andrew J Schaefer, Ellis L Johnson, Anton J Kleywegt, and George L Nemhauser. Airline crew scheduling under uncertainty. *Transportation science*, 39(3):340–348, 2005.

[31] Milind G Sohoni, Ellis L Johnson, and T Glenn Bailey. Operational airline reserve crew planning. *Journal of Scheduling*, 9(3):203–221, 2006.

[32] Goran Stojković, François Soumis, Jacques Desrosiers, and Marius M Solomon. An optimization model for a real-time flight scheduling problem. *Transportation Research Part A: Policy and Practice*, 36(9):779–788, 2002.

[33] Mirela Stojković, Francois Soumis, and Jacques Desrosiers. The operational airline crew scheduling problem. *Transportation Science*, 32(3):232–245, 1998.

[34] Pamela H Vance, Cynthia Barnhart, Ellis L Johnson, and George L Nemhauser. Airline crew scheduling: A new formulation and decomposition algorithm. *Operations Research*, 45(2):188–200, 1997.

[35] Megan Wiggins and Kevin Conrad. Next day crew scheduling at Jazz Aviation. Technical report, Dalhousie University, Halifax, Nova Scotia, Canada, April 2014.

[36] Shangyao Yan and Yu-Ping Tu. A network model for airline cabin crew scheduling. *European Journal of Operational Research*, 140(3):531–540, 2002.

Appendix A

Local and UTC Times and Dates

Depending on the direction of flight, i.e., how time zones change as a flight takes place from origin towards destination, the local arrival date at the destination may be the same as the local departure date at the origin or one calendar date apart. Clearly this is true under the assumption that typical aircrafts that are used to fly passengers in this day are not normally scheduled to fly non-stop for 24 hours in a row or longer. Given the local departure date, local departure time, local arrival time, departure airport UTC offset, and arrival airport UTC offset of any leg, the corresponding local arrival date is determined using the following pseudo-code:

START.

%Case 1 (Flight does not pass midnight):

if Local Arrival Time $\geq$ Local Departure Time:

    then: Local Arrival Date $\leftarrow$Local Departure Date;

%Case 2 (If flying from east to west; in North America this translates to flying towards a more negative UTC offset, i.e., arrival time zone has a more negative UTC offset than departure time zone):

else if (Local Arrival Time $<$Local Departure Time) **and**

(absolute value of [Departure City UTC Offset] $<$absolute value of [Arrival City UTC offset]):

    then: Local Arrival Date $\leftarrow$Local Departure Date;

%Case 3 (If flying from west to east or within the same time zone; in North America, this is equivalent to departure time zone having the same or a more negative UTC offset than arrival time zone):

else if (Local Arrival Time $<$Local Departure Time) **and**

absolute value of [Departure City UTC Offset] $\geq$ absolute value of [Arrival City UTC offset]:

then: Local Arrival Date ←date following Local Departure Date.

END.

Case 1 is the most common case encountered in the datasets provided by the airline, while cases 2 and 3 are seen occasionally. An example of each case is given in Table A.1. To illustrate how the pseudo-code presented above works, the arrival date

Instance	Case	Flight No.	Origin Airport	Destination Airport	Departure Date	Local Departure Time	Local Arrival Time
1	Case 1	7061	YUL	YAM	09/01/15	0800	1025
2	Case 2	8721	YGR	YGP	05/29/15	1450	1438
3	Case 3	7066	YYC	YWG	09/01/16	2100	0023

Table A.1: Three different cases encountered when calculating the local arrival date of a leg

of the sample leg corresponding to Case 2 in Table A.1 will be determined. For this purpose, the UTC offsets of the origin and destination airports are required and can be found in Table A.2.

Airport	Minimum UTC Offset	Maximum UTC Offset	Local Time	Local Date
YGR	-04:00	-03:00	14:50	05/29/15
YGP	-05:00	-04:00	14:38	Local Arrival Date = ?

Table A.2: Example of local arrival date calculations

In this example, based on the time of year which calls for Summer Time calculations, Minimum UTC Offsets are used (as opposed to Winter Time/Maximum UTC Offsets). Therefore, the time zones corresponding to departure and arrival cities YGR (with an offset of -04:00) and YGP (with an offset of -05:00), show that the direction of flight is from east to west and therefore since the local arrival time is less than, i.e., earlier than, the local departure time, the correct local arrival date is the same as the local departure date, namely 05/29/15. The verification of cases 1 and 3 are left to the reader.

Appendix B

List of Parameters and Variables Used

In this section, the parameters and variables used in equations throughout this thesis with regards to duty periods and pairings are listed.

Constant values obtained from CBA:

- MIN-SIT
- DH-IN-BUFFER
- DH-OUT-BUFFER
- $SilentHoursStartTime$
- $SilentHoursEndTime$
- CHECK-IN-TIME
- CHECK-OUT-TIME
- MAX-LENGTH-OF-DUTY
- MIN-LENGTH-OF-LAYOVER
- MAX-NUMBER-OF-LEGS-IN-DUTY
- MAX-NUMBER-OF-DUTIES
- MIN-GUARANTEED-CREDIT-HRS-PER-DUTY

Variables:

- DutyCheckInTime

- DutyCheckOutTime
- Duty-Elapsed-Time
- PairingCheckInTime
- PairingCheckOutTime
- Pairing-Elapsed-Time
- Duty-Block-Hours
- Duty-Credit-Hours
- Pairing-Block-Hours
- Pairing-Credit-Hours

Parameters used in Experiment described in Section 4.2.1:

- MAX-LENGTH-OF-SIT
- Longest-Sit-Found
- 2nd-Longest-Sit-Found

Appendix C

GMPL Code for Phase 2

The pairing optimization problem modeled in GMPL is as follows. In Gusek, GLPK solver can easily be used to solve regular-sized problems.

```
#---------------------------------------------------------------#
#                       PAIRING OPTIMIZATION CODE               #
#---------------------------------------------------------------#

set L;

param e{l in L};
/* parameters */

table tab_parameters IN "CSV" "parameters.csv" :
 L <- [parameter], e ~ value ;

param m, integer := e['m'];
/* index of last open leg */
/* = number of open legs */

param n, integer := e['n'];
/* index of last feasible pairing */
/* = number of feasible pairings minus 1 */

set I := {1..m};
/* set of open legs */

set J := {0..n};
```

```
/* set of feasible pairings */


param a{i in I, j in J}, default 0;
/* whether open leg i is on pairing j; */
/* OUTPUT OF PHASE 1*/

param bh{j in J};
/* block hour cost of feasible pairing j; */
/* not necessarily integer; naturally ch[j] >= bh[j] */

param ch{j in J};
/* credit hour cost of feasible pairing j; */
/* not necessarily integer */

param ln{j in J};
/* length of feasible pairing j (in hours); */
/* not necessarily integer */

table tab_a_matrix IN "CSV" "a_matrix.csv" :
 [OpenLeg, ValidPairing], a ~ a_matrix_coefficient;
/* constraint matrix, where the rows represent open legs */
/* and the columns represent feasible pairings */

table tab_bh_matrix IN "CSV" "bh_matrix.csv" :
 [ValidPairing], bh ~ bh_matrix_coefficient;
/* block hour cost of feasible pairing j */

table tab_ch_matrix IN "CSV" "ch_matrix.csv" :
 [ValidPairing], ch ~ ch_matrix_coefficient;
```

```
/* credit hour cost of feasible pairing j */

table tab_ln_matrix IN "CSV" "ln_matrix.csv" :
 [ValidPairing], ln ~ ln_matrix_coefficient;
/* length of feasible pairing j (in hours) */
/* from pairing check-in time to pairing check-out time */

var x{j in J}, binary;
/* decision variables; x[j] = 1 means feasible pairing j is selected */

minimize creditLoss: sum{j in J} ((ch[j] - bh[j]) * x[j]) +
                     0.000001*sum{j in J} (ln[j] * x[j]);
/* the objective is to find the cheapest selection of feasible pairings
that cover each open leg exactly once */
/* in case of a tie, preference is given to the pairing with less
deadheading and thus shorter length */

s.t. legCovering{i in I}: sum{j in J} (a[i,j] * x[j]) = 1;
/* constraints; each open leg must be covered by exactly one pairing; */
/* note that the same flight leg which is in open-time for more than */
/* one crew position is given a unique leg identification number, and thus */
/* every combination of leg/position is considered a separate open segment */

solve;

table tab_result{j in J} OUT "CSV" "result.csv" :
 j ~ ValidPairing, x[j] ~ Selection;

end;
```

Appendix D

MPL Code for Phase 2

The pairing optimization problem modeled in MPL is as follows. The Gurobi Solver can conveniently be used to solve large problems.

```
!---------------------------------------------------------------!
!                       PAIRING OPTIMIZATION CODE               !
!---------------------------------------------------------------!

{ValidPairingOptimization.mpl}


TITLE

    ValidPairingOptimization;


OPTIONS

   ExcelWorkbook = "parameters.xlsx";


DATA

    M := EXCELRANGE("m");
    N := EXCELRANGE("n");
```

```
INDEX

  i = 1..M;
  j = 0..N;

DATA

    BH[j]  := DATAFILE("bh_matrix.dat");
    CH[j]  := DATAFILE("ch_matrix.dat");
    LNG[j] := DATAFILE("ln_matrix.dat");
    A[j,i] := SPARSEFILE("a_matrix.dat");

BINARY VARIABLES

    x[j];

MODEL

    MIN  CreditLoss =
  SUM(j: CH[j]*x[j] -BH[j]*x[j]) + 0.000001*SUM(j: LNG[j]*x[j]);


SUBJECT TO

    LegCovering[i] : SUM(j: A[j,i]*x[j])  =  1 ;

END
```

Appendix E

Uncovered Segments Prediction

The following flowchart shows how to predict whether a particular open leg will be left in open-time with regards to a single crew base. Clearly, in a multi-base setting, if a leg cannot be covered via any of the pairings belonging to different bases, then it would certainly be left in open-time. Note that the terms *predecessor*/*successor* are used to refer to open legs (from the entire set of open legs given as input) that can precede/follow the segment under consideration in a pairing in terms of location and minimum sit connection time (i.e., logical constraints). In Figure E.1, *DH* refers to deadhead segments.

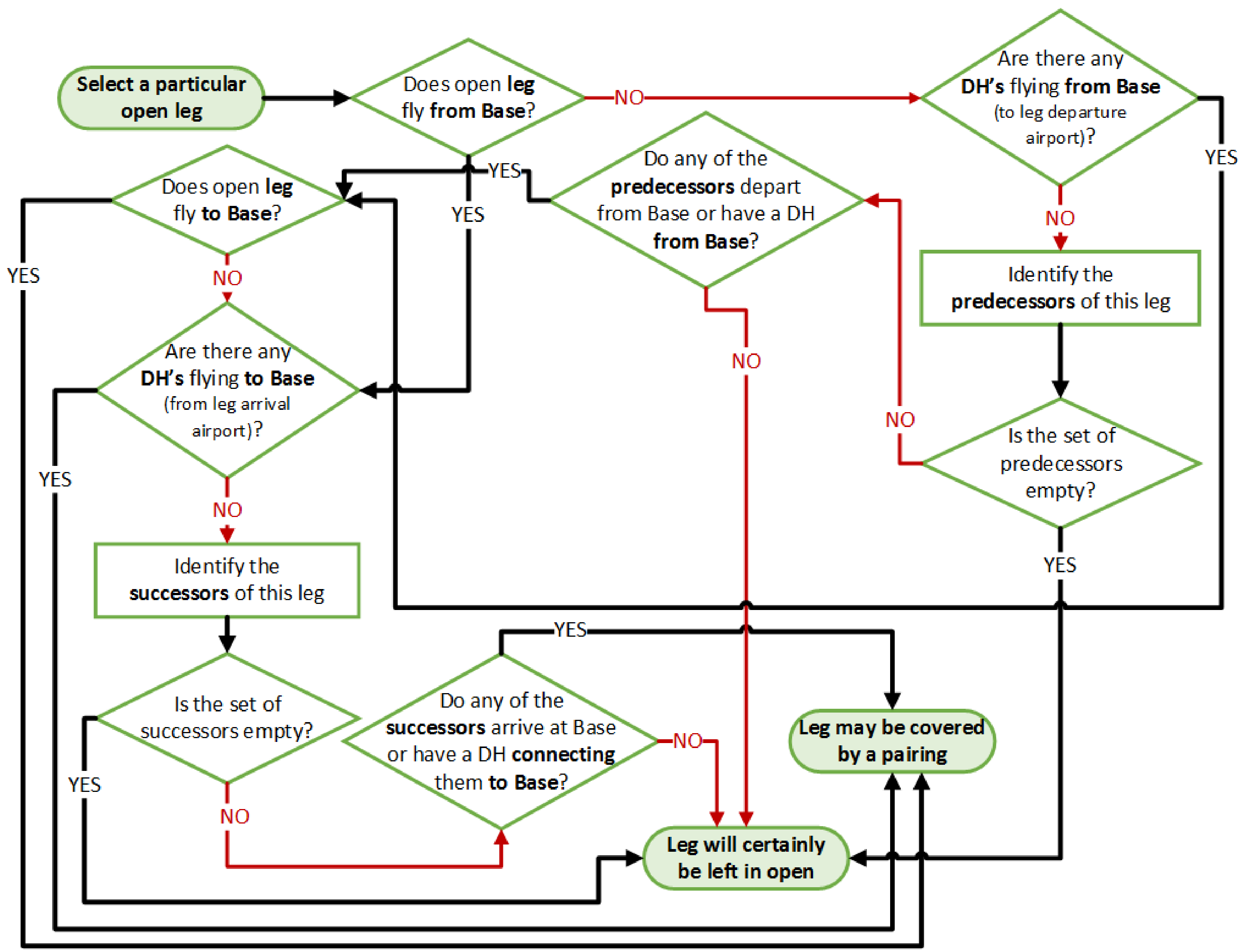


Figure E.1: Flowchart predicting whether a particular open leg would be left in open