# University of Thessaly

Department of Mechanical Engineering


# Diploma Thesis

## Incorporation of Min-Max Constraints in Branch and Price Methodologies for Airline Crew Scheduling


Author

Spyridon-Tsarls Botsis


Submitted in partial fulfilment of the requirements for the post graduate Diploma of Supply Chain Management & Logistics from University of Thessaly.

Volos, February 2023

**Approved by the Members of the Examination Committee:**

First Examiner: Dr. George Kozanidis
(Supervisor) Associate Professor, Department of Mechanical Engineering, University of Thessaly

Second Examiner: Dr. George Liberopoulos
Professor, Department of Mechanical Engineering, University of Thessaly

Third Examiner: Dr. Georgios Saharidis
Associate Professor, Department of Mechanical Engineering, University of Thessaly

# Abstract

This diploma thesis investigates the Crew Rostering Problem encountered in the context of airline management. The problem is typically tackled with two alternative formulations, the set cover and the set partition optimization model. The first formulation aims to successfully cover all the pairings with the given set of crew members without taking into consideration roster quality. The set partition model, on the other hand, tries to successfully cover all the pairings with the given set of crew members while also respecting the exact complement of each flight duty, as well as optimizing roster quality. To both these formulations, we add a special set of constraints imposing a strict upper limit on the number of crew members with a special characteristic assigned to specific flight duties. A branch and price solution methodology is employed for reaching an optimal solution in reasonable computational time. The proposed methodology is initiated with a master problem in which every crew member is assigned an empty roster while all pairings remain uncovered. A column generation is successfully utilized for adding promising rosters to the master problem, based on their reduced cost with respect to its linear relaxation optimal solution. This procedure also ensures that the generated rosters respect certain rules based on the regulation scheme imposed by the airline administration. To evaluate the proposed methodology and the effect of these new constraints, we conduct computational experiments with realistic scenarios and we discuss the obtained results. The proposed methodology can assist airline practitioners to satisfy special constraints the crew member rosters must abide with.

# Acknowledgments

I would like to thank my MSc Thesis Advisor Dr. George Kozanidis for his consistent and high-level guidance provided throughout the fulfilment of the present MSc thesis. I feel very thankful for his help in making a significant contribution towards my dissertation. The understanding, patience and professionalism demonstrated were very much appreciated. I am very grateful for his guidance and wisdom, which made me grow in both knowledge and professional experience. This thesis would not be possible without his efforts in assisting me to make an important contribution in the field of supply chain management.

I would also like to thank my family for the support and understanding they have shown me throughout my studies. Without their encouragement and patience, I would not have been able to complete my education. I am especially grateful to my parents, who have always believed in me and helped me reach my goals. I know that their proud of me and that makes me very glad.

I would also like to express my gratitude to all the committee members for showing interest in my research and playing an important role in my fulfillment in the requirements of post graduate degree, Dr. George Liberopoulos and Dr. George Saharidis.

# Table of Contents

1 Introduction ........ 1
1.1 Thesis outline ........ 1
1.2 The History of the Airline Industry and How it's Evolved Over Time ........ 2
1.2.1 Early Days and the Beginning of Commercial Passenger Travel ........ 2
1.2.2 Economic Deregulation ........ 3
1.2.3 The present ........ 4
1.3 Operational Research in the aviation industry ........ 7
1.3.1 Revenue Management ........ 7
1.3.2 Applications to Aviation Infrastructure ........ 8
1.3.3 Aircraft and Crew Schedule Planning ........ 9
1.4 Thesis objectives and motivation ........ 11
2 Literature Review ........ 13
2.1 Crew Pairing ........ 13
2.2 Crew Rostering ........ 14
2.3 Crew Rostering with Min-Max Constraints ........ 16
3 Model Formulation and Solution Methodology ........ 18
3.1 Master Problem formulation ........ 19
3.1.1 Set Partition ........ 19
3.1.2 Set Cover ........ 21
3.1.3 Adjusted models with Minimum or Maximum Constraints ........ 22
3.2 Branch and Price methodology ........ 23
3.2.1 Column Generation ........ 23
3.2.2 Branch and Bound ........ 27
3.3 An overview of the methodology with an Example ........ 29
3.3.1 Overview of Branch and Price methodology ........ 29
3.3.2 Example case study ........ 30
4 Computational Study ........ 43
4.1 Setting the Scenarios ........ 43

4.2 Results ........................................................................................................ 44
4.2.1 Scenario 1 – Balanced and low utilization.......................................................... 44
4.2.2 Scenario 2 - Balanced and high utilization .......................................................... 45
4.2.3 Scenario 3 - Unbalanced and low utilization ........................................................ 47
4.2.4 Scenario 4 - Unbalanced and high utilization ....................................................... 48
4.2.5 Scenario 5 – Flexible and low utilization ............................................................. 49
4.2.6 Scenario 6 – Flexible and high utilization ............................................................ 50
5 Summary........................................................................................................ 51
5.1 Future work ..................................................................................................... 51
6 Bibliography .................................................................................................... 53

**Table of Figures**

Figure 1.1 Hub and Spoke networks [3]. .... 4
Figure 1.2 Total Passenger Traffic [3]. .... 5
Figure 1.3 Evolution of the average price of air travel [4]. .... 6
Figure 1.4 International tourist arrivals by mode of transport, 2017 [3]. .... 6
Figure 1.5 Airline Revenue Management System [4]. .... 8
Figure 1.6 Schedule Planning [7]. .... 10
Figure 2.1 Examples of horizontal rules and costs [23]. .... 15
Figure 2.2 Crew Rostering Problem [8]. .... 16
Figure 3.1 Set Partition Formulation. .... 19
Figure 3.2 Set Cover Formulation. .... 21
Figure 3.3 Column Generation flowchart [26]. .... 24
Figure 3.4 Network G [26]. .... 25
Figure 3.5 An illegal connection between the nodes [26]. .... 26
Figure 3.6 Branching. .... 28
Figure 3.7 Branch and Price flowchart [27]. .... 30
Figure 3.8 Beginning Restricted Master Problem. .... 32
Figure 3.9 Dual of Master restricted problem. .... 33
Figure 3.10 Rosters added to RMP for Crew Member 0. .... 33
Figure 3.11 Rosters added to RMP for Crew Member 1. .... 34
Figure 3.12 Rosters added to RMP for Crew Member 2 & 3. .... 35
Figure 3.13 Rosters added to RMP for Crew Member 0 in second iteration. .... 36
Figure 3.14 All rosters generated by the Column Generation algorithm. .... 37
Figure 3.15 Branching on variable x50. .... 38
Figure 3.16 RME after branching in Child Problem 2. .... 39
Figure 3.17 RME after branching in Child Problem 1. .... 40
Figure 3.18 RME after second complete iteration of Column Generation. .... 41
Figure 3.19 Finding an integral solution. .... 42
Figure 4.1 Time Comparison plot for scenario 1. .... 45
Figure 4.2 Time Comparison plot for scenario 2. .... 46
Figure 4.3 Time Comparison plot for scenario 3. .... 47

Figure 4.4 Time Comparison plot for scenario 4. ........................................................................ 48
Figure 4.5 Time Comparison plot for scenario 5. ........................................................................ 49
Figure 4.6 Time Comparison plot for scenario 6. ........................................................................ 50

**List of Tables**

Table 3.1 Crew Members Data ........................................................................................ 30
Table 3.2 Pairing Data ........................................................................................ 31
Table 4.1 Scenario 1 Parameters. ........................................................................................ 44
Table 4.2 Scenario 1 Computational results. ........................................................................................ 44
Table 4.3 Scenario 2 Parameters. ........................................................................................ 45
Table 4.4 Scenario 2 Computational results. ........................................................................................ 46
Table 4.5 Scenario 3 Parameters. ........................................................................................ 47
Table 4.6 Scenario 3 Computational results. ........................................................................................ 47
Table 4.7 Scenario 4 Parameters. ........................................................................................ 48
Table 4.8 Scenario 4 Computational results. ........................................................................................ 48
Table 4.9 Scenario 5 Parameters. ........................................................................................ 49
Table 4.10 Scenario 5 Computational results. ........................................................................................ 49
Table 4.11 Scenario 6 Parameters. ........................................................................................ 50
Table 4.12 Scenario 6 Computational results. ........................................................................................ 50

# 1 Introduction

## 1.1 Thesis outline

This thesis consists of a qualitative introduction and detailed chapters to fully understand the formulated models and the followed methodologies.

Starting with the introduction, after a short outline of the thesis, a history from the beginning of the airline industry till the complex and big scale operations followed today is presented. Through the time we will see the operational issues which came with the growing of the industry and the solutions provided through the years to tackle those issues. Finishing this chapter, we will present the research scope of this thesis and the motivation behind it.

In Chapter two, we will do an extensive literature review of the existing methodologies developed to tackle issues faced in the airline industry as well as similar implementations which we can use as motivation.

In chapter three, the formulation of the models will be presented for the crew rostering problem. To those models, min-max constraints will be introduced to fulfil different kind of requirements not captured by the basic formulations. Next, a branch and price solution methodology will be employed for reaching a satisfactory solution in reasonable time. Then, an example case study will be given to illustrate the application of the method.

In the fourth chapter, the new proposed formulation will be tested in order to evaluate the solutions it provides and its computational requirements. Then, the results will be compared to the original formulation in order to assess the impact of the min-max constraints.

At the fifth chapter, a to the point summary will be presented with key conclusions and possible future steps for further research.

The final chapter will have the bibliography.

## 1.2 The History of the Airline Industry and How it's Evolved Over Time

The airline industry is one of the most important industries in the world. It has been a major player for the past century and it continues to grow. That being said, it is important to know the history of this industry in order to understand how we got here and what might happen in the future. This section will provide a brief overview of the milestones, challenges, and innovations that have shaped this industry. The sections to follow are heavily influenced by [1]

### 1.2.1 Early Days and the Beginning of Commercial Passenger Travel

The invention of the airplane is a major breakthrough in aviation. It has allowed people to travel long distances in short time without needed effort to develop infrastructure such as roads and rails. The first airplane was invented by the Wright Brothers who built and flew their first successful airplane in 1903. Their plane had no propeller and was powered by an engine that ran on gasoline. The flight lasted only 12 seconds covering a distance of 37 meters, the first powered flight in a heavier-than-air machine. This invention was patented with U.S Patent Office on May 22, 1906.

Due to the general public being afraid to travel with the newly developed airplanes, commercial aviation was very slow to catch on. This also delayed the improvements in the design and manufacturing. However, with the beginning of World War I, the militaristic benefits provided by the aircrafts quickly ramped up the production and development. The high demand for improvements gave forth to the significant improvement of the aircraft's engines. This enabled the aircraft to reach speeds up to 210 kilometres per hour. The new powerful engines allowed the development of larger aircrafts.

With the end of WWI, the demand of aircraft plummeted while there was a large surplus of already existing planes. The commercial aviation was not in a better spotlight as before the war as the public had correlated the airplanes as more as a weapon for destruction than a tool for society. Despite that, in some European countries, such as Great Britain and France, it was being used as trips were frequent over the English Channel. In the U.S.A., commercial aviation was mostly a government program, the transportation of mail by air.

The joint efforts of the army and the post office to transport the mail started with small distance flights between states but quickly the ambitions grew larger. With the large surplus of aircraft from the war, the Post Office sets the goal for transcontinental air service and soon after achieved to deliver mail coast to coast.

In 1927 an event brought great public attention to the aviation industry; Charles Lindbergh made a solo flight across the Atlantic Ocean. The historic flight started from New York and reached Paris, it lasted 33 hours, 29 minutes, and 30 seconds no stop. This was a great achievement which attracted a great interest from investors and the public.

Other technological advancements had an enormous impact on the expansions of the aviation industry like the radio. The radio was used to transmit weather information to the pilots so they can alter their course to avoid storms and in general bad weather. This was the reason for the creation of the first air traffic control tower in 1935.

The first aircraft to be considered in a modern airline was built in 1933 by Boeing. The Boeing Model 247 could accommodate ten passengers and fly at a rate of 155 miles per hour. The cabin was insulated to reduce the noise produced by the engines and provided additional amenities like hot water for beverages and upholstered seats. Soon after, Douglas Aircraft Company created an aircraft which incorporated and improved on many of Boeing's innovations. The aircraft was named DC-1 but only one was created as it was almost immediately decided to go to improvements and design DC-2, which was a great success.

The next breakthrough was the pressurized cabins as previously aircraft could only fly up to 3 thousand meters. Going higher than that limit caused the people to become dizzy and even faint due to lower levels of oxygen in those altitudes. Flying up to that altitude also meant that the aircraft could not go over the weather resulting in turbulence of the flight and rerouting was needed to avoid obstacles like high mountains and storms. By pumping air in the aircraft's cabin, the air pressure remained stable while the aircraft gained altitude up to 7 thousand meters overcoming those challenges.

With the invasion of Poland by Hitler's Germany in 1939, the second world war started. The importance of the aviation industry in the war efforts was great and provides a significant military advantage. The research in aviation to gain a significant advantage over their enemies resulted in mass production of aircraft, the development of the radar, and the development of the first jet engine. After the WWII the commercial aircraft industry incorporated the new developments in the new aircraft making them bigger, faster, cheaper, and more reliable. The beginning of the Jet age began in 1958 with Boeing 707, this aircraft could reach speeds of 880 kilometres per hour and carry 181 passengers. After 20 years, another revolutionary aircraft made its debut, Boeing 747. The Boeing 747 was the first wide body jet with 2 decks, able to carry up to 450 passengers. This aircraft was 80% larger than the other developed jets, this enabled the economy of scale which was a major disruption for the airline industry at that time.

### 1.2.2 Economic Deregulation

About seventy years after the development of the airplanes, an airplane was able to transport hundreds of people fast, safe, and comfortable over continents in a single flight. The technology had advanced so significantly, but they were other obstacles in the further growth of the commercial airline industry.

Over the world, the airline industry was restricted by government agencies, the routes and fares were determined by them, thus the industry resembled a public utility. Pressure for airline deregulation was slowly building up. Many economists pointed out that the liberalization of the air transport market would largely benefit passengers and the airline companies. In addition, the introduction of the wide-body airplanes, which increased the capacity of a flight, made it hard to cover the extra seats in the market without adjusting the fares. Also, in 1973 there was an embargo on Arab oil which surged the fuel costs affecting the fare rates.

These and other events lead U.S.A. to the 1978 Deregulation Act, which stated that the route and rate restrictions imposed will be steadily eliminated. This sparked a change all over the globe such as in Europe, with a decade long process starting in 1986. Several sets of EU regulations gradually turned the once protected national aviation markets into a competitive single market for air transport [2].

The result of the deregulation had an impressive effect on the industry. In order to serve efficiently bigger markets with a given fleet, airlines developed a strategy called Hub and Spoke network. The Hub and Spoke networks work with a batch of flights arriving at a single airport to leave shortly affect, giving the opportunity to passengers or cargo to board another flight to a different destination. This gave the opportunity to passengers to easily fly to destinations which otherwise would not be possible as they were no direct flights available. This also resulted in higher load factors even for flights to airport of smaller cities which could not be operated profitably previously.

Further deregulation offered the ability for new airlines to be established, increasing the fare competition. Thus, the passengers experienced a great benefit of discounted prices and ability to choose desired comforts. The opening of the airline market brought a remarkable increase of operating companies, for example in the U.S.A in 1978 they were around forty airline carries and in 2005 the number was increased roughly by one hundred. To differentiate from the competitors, the airlines operated different business models, the most notable being the low-cost model.

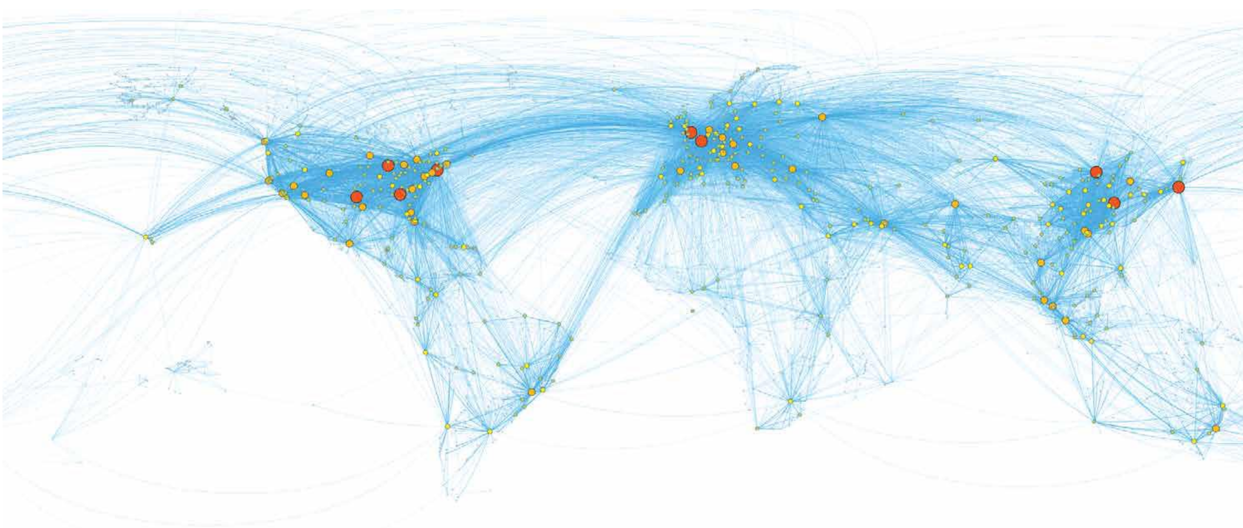

*Figure 1.1 Hub and Spoke networks [3].*

### 1.2.3 The present

Aviation has continued to expand. It has weathered crises and demonstrated long-term resilience, becoming an indispensable means of transport. Historically, air transport has doubled in size every fifteen years and has grown faster than most other industries. In 2018, airlines worldwide carried around 4.3 billion passengers annually with 8.3 trillion revenue passenger kilometres (RPKs). Fifty-eight million tonnes of freight were transported by air, reaching 231 billion freight tonne kilometres (FTKs). Every day, more than 100,000 flights transport almost 12 million passengers and around USD 18 billion worth of goods [3].

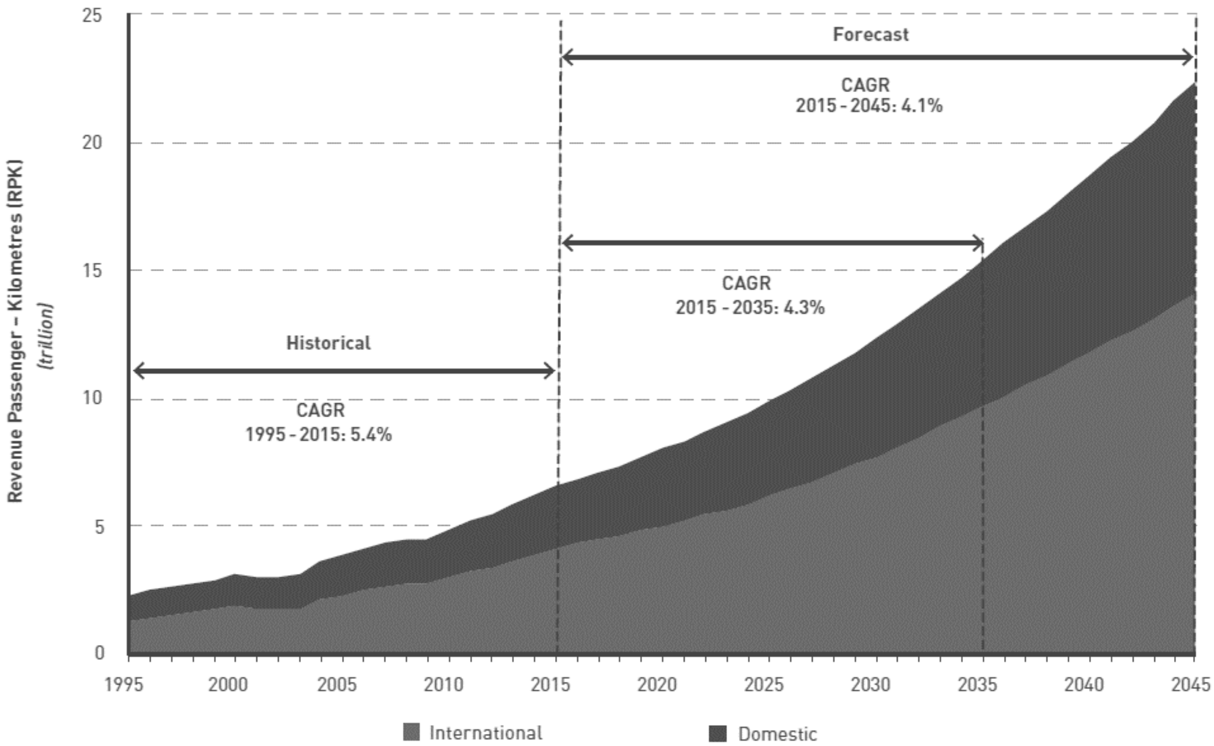


*Figure 1.2 Total Passenger Traffic [3].*

With the great scale and competition of the today's airliners, the air travel has rapidly grown covering millions of flights over the globe annually. The different models followed by the airlines have brought the average fare rate to the lowest from the beginning of the airline history, as seen in Figure 1.3.

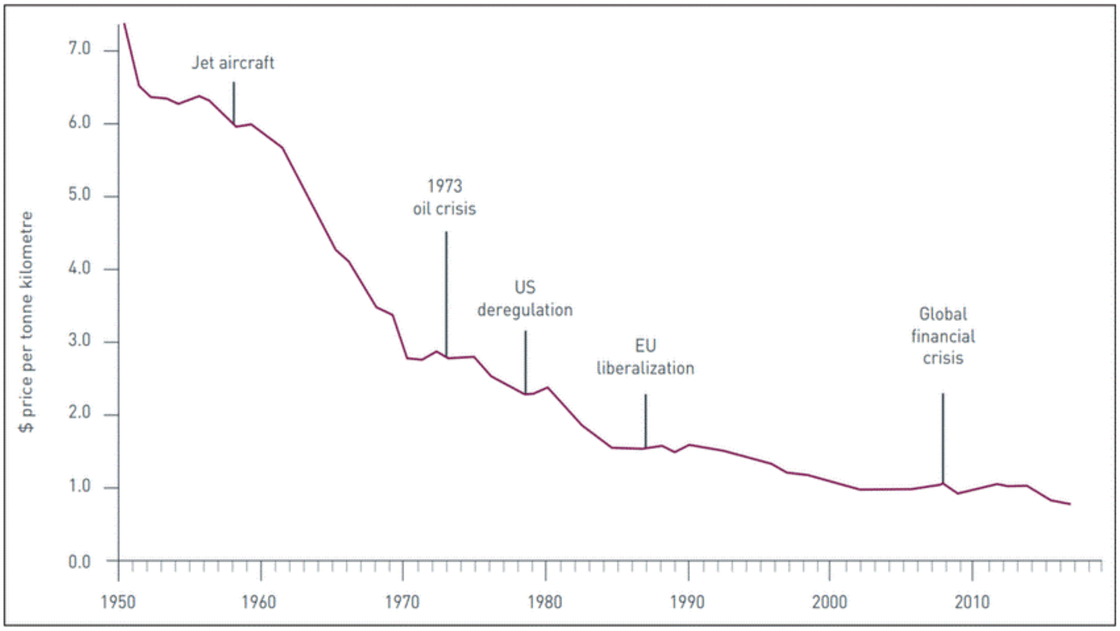


*Figure 1.3 Evolution of the average price of air travel [4].*

Further than the direct benefits of the aviation industry, they are other industries heavily dependent on it like tourism where research shows that more than 50 percent of tourists arrive at their destinations by air as seen in Figure 1.4. This brings economic growth over regions which are not capable of developing other industries alleviating poverty. It also brings different cultures together which enriches the relationship between nations.

In addition, supply chains have incorporated air transport in their streams due to the fast and reliable delivery worldwide. Using this benefit, companies may use the technological advantages already set up in many other countries for some processes they follow and fly their unfinished products for menial intensive procedures to countries with lower labour cost.

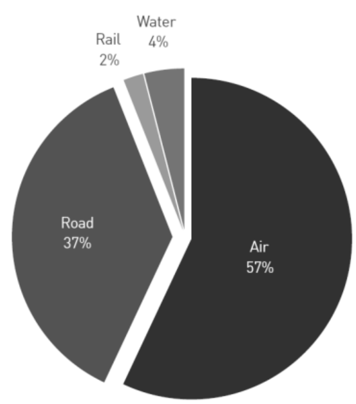


*Figure 1.4 International tourist arrivals by mode of transport, 2017 [3].*

## 1.3 Operational Research in the aviation industry

As stated in [5], the reference on which this subchapter is based on, Operations Research has been one of the principal contributors to the enormous growth that the air transport sector has experienced during the past 50 years. In the best tradition of Operations Research, the development of models and solutions has been motivated by issues and problems encountered in practice and has led, in several instances, to insights of a general nature and to important methodological advances in the Operations Research field at large. At this point, Operations Research models and algorithms are diffused throughout the sector and constitute an integral part of the standard practices of airlines, airports, and Air Traffic Management service providers. In view of the numerous challenges that it currently faces, it is safe to expect a continuing central role for Operations Research in the air transport sector’s future.

There are many relevant topics of research, such as:

- Revenue Management
- Applications to Aviation Infrastructure
- Aircraft and Crew Schedule Planning

### 1.3.1 Revenue Management

It is common to have some flights with high demand which cannot be covered by the available seats, and some other ones which operate with empty seats. To better match the demand of the flight with the aircraft’s capacity and to increase total revenue, the airlines offer a large variety of fare prices and comforts for each flight. To determine the fare level of each seat and maximize the possible revenue, Revenue Management is used. Revenue Management is heavily dependent on Information technology due to the sheer size and complexity of the airlines. With even a medium sized airline, 1000 flights might be daily operated with 10 fare levels and booking up to a year prior the departure. It is only possible to optimally handle this situation with the assistance of IT systems, since the medium size airline's seat inventory includes over three million booking limits at any given time, which can change with each accepted booking. A forecast of the demand of each fare class of a flight is produced by a combination of previous bookings, the day of the week, and real time reservations for future flights.

These forecasts, together with estimates of the revenue value of each booking class, are then fed into an optimization model that calculates the recommended booking limits for the flight departure in question. At the same time, the demand forecasts are fed into an overbooking model, which also makes use of historical information about passenger no-show rates to calculate an optimal overbooking level for the future flight departure. Both the booking class limits and overbooking levels calculated by the mathematical models are then presented as recommendations to the Revenue Management analyst.

The demand forecasts and booking limits are reviewed by the Revenue Management system at regular intervals during the flight booking process, as often as daily in some cases. Should unexpected booking activity occur, the system reforecasts demand and reoptimizes its booking limit recommendations. A substantial proportion of the revenue gain attributable to fare mix optimization comes from this dynamic revision of booking limits.

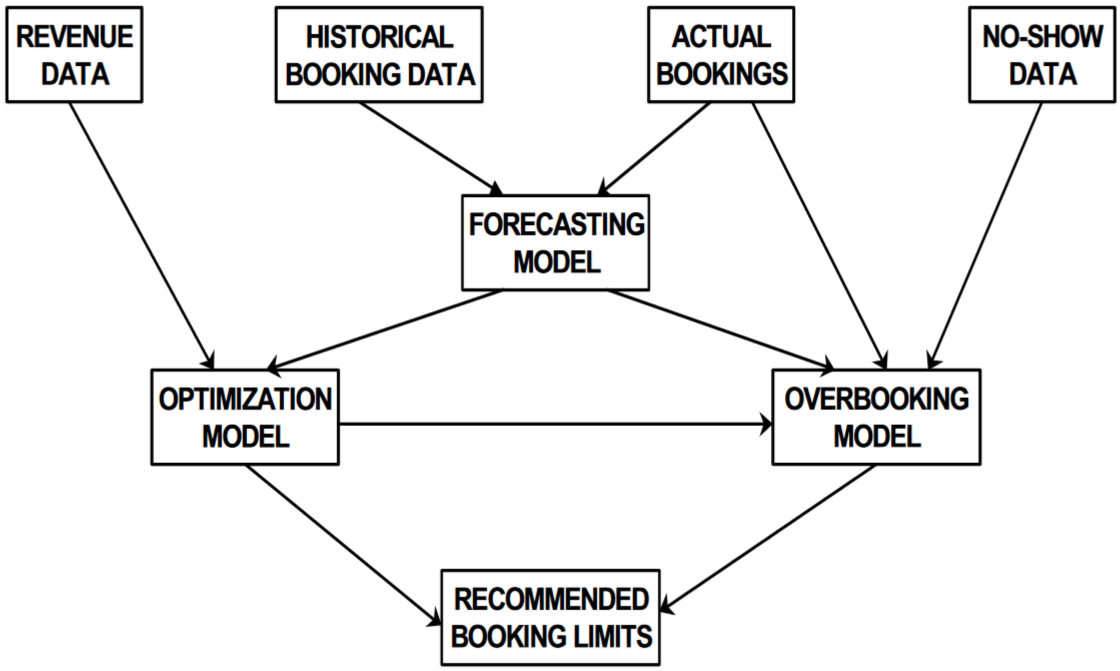


*Figure 1.5 Airline Revenue Management System [4].*

Most large and medium-sized airlines throughout the world have implemented third-generation Revenue Management systems, and the benefits of such systems have been well documented. Effective use of techniques for overbooking and fare class mix alone have been estimated to generate revenue increases of as much as 4%–6% compared to situations in which no seat inventory control tools were applied [6].

### 1.3.2 Applications to Aviation Infrastructure

The global aviation systems' infrastructure has two key elements, airports and air traffic management (ATM) systems. Airports may be further divided into airside facilities, such as runaways, aircraft stands, and landside facilities such as passenger infrastructure and cargo infrastructure. On the other hand, air traffic management is split in air traffic control and air traffic flow management. These systems gained great interest of operations research from the 60s-70s where the aviation experienced its greatest growth and the efficient management of these key elements became apparent.

The Airport Operations consists of many aspects, but the most important one is the runway complexity of major airports. New runways are very expensive to build and require a large area to be constructed, and further than that, have a great impact on the near environment and need a long and complicated assessment processes. This makes them a scarce resource in the international air transport system and the efficient use of such key resource received much attention from the area of transportation science. Results of such attentions are airside simulations, optimizations of airside operations and analytical capacity and delay models.

The most advanced research in the aviation infrastructure has been applied to the Air Traffic Flow Management (ATFM). The ATFM was made necessary in the 80s due the rapid traffic growth. The goal of the ATFM is to prevent overloading in local systems by adjusting dynamically the flow of the aircraft in a great region. It develops flow plans that attempt to dynamically match traffic demand with available capacity over longer time horizons, typically extending from 3–12 hours in the future. The prototypical application of ATFM is in ground holding, i.e., in intentionally delaying an aircraft’s take off for a specified amount of time to avoid airborne delays and excessive controller workload later on. Other ATFM tactics include rerouting of aircraft and controlling the rate of traffic flows through specified spatial boundaries in airspace [5].

### 1.3.3 Aircraft and Crew Schedule Planning

Schedule planning involves designing future aircraft and crew schedules to maximize airline profitability. This problem poses daunting challenges because it is characterized by numerous complexities, including a network of flights, differing aircraft types, gate, airport slot and air traffic control restrictions, noise curfews, maintenance requirements, crew work rules, and competitive, dynamic environments in which passenger demands are uncertain and pricing strategies are complex. Not surprisingly, no single optimization model has been solved, or even formulated, to address this complex design task in its entirety. The problem’s unmanageable size and complexity has resulted in the decomposition of the overall problem into a set of subproblems, often defined as follows:

1. *Schedule design*: Defining which markets to serve and with what frequency, and how to schedule flights to meet these frequencies.
2. *Fleet assignment*: Specifying what size aircraft to assign to each flight.
3. *Aircraft maintenance routing*: Determining how to route aircraft to ensure satisfaction of maintenance requirements.
4. *Crew scheduling*: Selecting which crews to assign to each flight to minimize crew costs.

Suboptimal, yet feasible aircraft and crew plans are constructed by solving the subproblems in order, constraining the solutions to subsequent problems based on the solutions to preceding problems. Although smaller and simpler than the overall problem, these subproblems are still large-scale and rich in complexity.

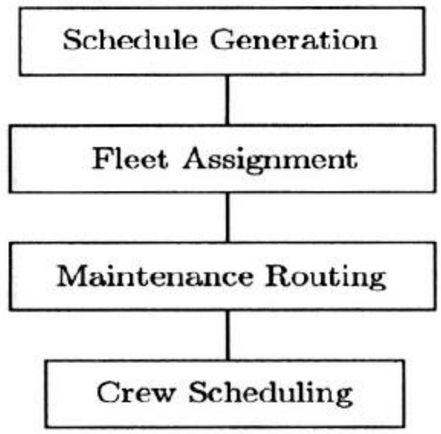


*Figure 1.6 Schedule Planning [7].*

**Schedule Design**

The flight schedule, specifying the flight legs to be flown and the departure time of each flight leg, largely defines the competitive position of an airline and is thus a key determinant of airline profitability. Designing a profit maximizing flight schedule, however, is extremely complex. It affects and is affected by essentially all aircraft and crew scheduling decisions of the airline and competing airlines as well. No single model has captured all these interdependencies and even if such a model were formulated, it surely would be intractable. Moreover, its input data requirements are impractical, requiring, for example, accurate estimates of itinerary-specific passenger demands, spill costs, and recapture rates.

Notwithstanding this complexity, flight schedule design and variants of the problem have been of interest to researchers for many years.

**Fleet Assignment**

With the flight schedule determined, the fleet assignment problem is to find the cost-minimizing assignment of aircraft types to legs in the flight network. Fleeting costs are comprised of:

1. Operating costs: Specified for each flight leg - aircraft type pair, representing the cost of flying that flight leg with that aircraft type.
2. Spill costs: Measuring the revenue lost when passenger demand for a flight leg exceeds the assigned aircraft's seating capacity.

**Aircraft Maintenance Routing**

With schedule design and fleet assignment decisions made, the flight network decomposes into subnetworks, each one associated with aircraft of a single type. The assignment of individual aircraft to flight legs in a subnetwork occurs in the aircraft maintenance routing step. The goal is to determine routings, or rotations, for each aircraft in a fleet. A routing is a sequence of flight legs, with the destination of one flight leg the same as the origin of the next leg in the sequence. A rotation is a routing that starts and ends at the same location. Each aircraft's rotation visits maintenance stations at regular intervals.

**Crew Scheduling**

After costs for fuel, crew costs constitute the second largest expense of an airline. Since profit is the difference between revenue and cost, cost efficient crew planning is of major importance for airlines. A cost reduction of a few percent usually results in annual savings of tens of millions for large airlines. Due to the potential for significant cost savings, operations research techniques were applied in the area of crew scheduling already at an early stage.

Due to its complexity, crew planning is usually divided into a crew pairing and a crew rostering phase. Firstly, anonymous pairings are formed out of the flight legs (flights without stop over) such that the crew needs on each flight are covered. So, a pairing is a sequence of legs to be assigned to one or more crew member working in one or more crew positions. The crew positions and the number of crew members a pairing must be assigned to is referred to as crew complement. Then in crew rostering, the pairings together with possible other activities such as ground duties, reserve duties and off-duty blocks are sequenced to rosters and assigned to individual crew members. Rostering is usually done one month at a time and each roster usually has some history. In both problems, complex rules and regulations coming from legislation and contractual agreements must be met by the solutions and some objective function must be optimized [8].

## 1.4 Thesis objectives and motivation

As stated, Operation Research has great interest in the aviation industry. The research areas are plenty with many challenging obstacles which need to be overcome.

Creating high quality crew schedules is of high importance for airlines as the crew expenditures is their second highest cost, just after fuel costs. In addition, the crew satisfaction plays a significic importance for the airlines, so crew preferences are taken into account for the scheduling process. Taking these into consideration, with the sheer size and complexity of the problem, a solution may be only provided by optimization techniques for large airlines.

Aircrew scheduling is usually performed in two sequential steps, crew pairing and crew rostering. In this thesis, the research scope focuses on the Crew Rostering Problem. Many restrictions need to be taken into consideration for creating a legal and optimal roster, with the involved checks typically being performed during the column generation process. Crew members of the same position, such as pilots, which are normally trained to operate a specific type of aircraft, are grouped per aircraft type and solved by the crew rostering problem. Other restrictions, which are not covered so thoroughly in the relevant literature, are cases such as language restrictions where a percentage of the cabin crew should be able to speak a specific language, or a specific percentage of the crew members need to have a specific characteristic or certification.

To address these restrictions, new constrains need to be introduced to the Crew Rostering Problem to increase the validity of the provided solutions and to increase the flexibility of the crew scheduling. These constraints will be set to force the rosters assigned to crew members with a specific characteristic of the solution to not overcome a specific limit per pairing.

# 2 Literature Review

As mentioned in the previous chapter, operation researchers have showed signific interest in the aviation industry and have heavily influenced the growth of it providing strategies which cut costs and improve utilization of its resource. The part which will be focused on is the crew sub problems crew pairing and crew rostering, while also we will investigate in more depth of the crew rostering with min-max constraints.

## 2.1 Crew Pairing

The Crew Pairing Problem is aimed to find a sequence of flight legs called pairings with the minimal cost. On any flight leg different positions of crew members are needed such as pilots, co-pilots, and cabin crew. Different rules apply to each category so that the crew pairing is normally decomposed to per category problems [9]. The formulation of the problem is based on a set of legs $F$ that must be operated during a given period. Let the set of all feasible pairings be $\Omega$, $a_{fp}$, $f \in F$, $p \in \Omega$ , be a constant equal to 1 if pairing $p$ contains leg $f$, and 0 otherwise, and $c_p$ be the cost of this pairing. For each $p \in \Omega$, define $x_p$ as a binary variable that takes value 1 if pairing $p$ is selected, and 0 otherwise. The Crew Pairing Problem is usually formulated as the following set-partitioning problem:

$$Min \quad \sum_{p\in\Omega} c_p x_p \tag{1}$$

$$s.t. \sum_{p\in\Omega} a_{fp} x_p = 1, \quad \forall f \in F \tag{2}$$

$$x_p \in \{0,1\}, \quad \forall p \in \Omega \tag{3}$$

The objective function (1) minimizes the total pairing costs. Constraints (2) ensure that each leg is covered exactly once, and constraints (3) enforce binary requirements on the pairing variables [10].

They are many other formulations which have additional problem constraints. The research in [11] includes a constraint which limits the total number of aircraft changes and the total number of deadheads. Another instance is [12], where the constraints aim to distribute evenly the work time amongst the crew bases. Most of these additional constraints are modeled as soft constraints with a penalization in the objective function.

For these problems, the number of feasible pairings is usually in the order of billions, even for small size instances. To address this issue, the Crew Pairing Problem is solved with column generation [13] [14]. This methodology is iterative and solves at each iteration a restricted master problem and one or several subproblems. The restricted master problem is the linear relaxation of the problem described above, which includes at each iteration only a small subset of all feasible pairings. The goal of the subproblem is to find negative reduced cost columns, for our case pairings, which improve the solution and are therefore added to the master problem. The column generation algorithm iterates between the restricted master problem and the subproblems till no other column with negative reduced cost for improving the solution can be identified.

The subproblem of the column generation algorithm for the crew pairing problem is formulated in most cases as a shortest path problem with recourse constraints (SPPRC) on an acyclic network. A major benefit of the network structure is that it accommodates a major part of the complex rules and cost structures. Two cases have been seen in most of the literature, the duty-based network, where each node corresponds to a duty and the arcs connect the duties that can be operated consecutively, and the flight-based network, where nodes represents to time-space coordinates and the arches to task performed by crew members. According to [15], although not as flexible as duty-based networks, the flight-based network is better suited to large Crew Pairing Instances with relative simple cost structure.

The column generation algorithm in general returns a fractional optimal solution; thus, a branch and bound algorithm is applied subsequently for obtaining an integer solution. By applying column generation on each branching node, its linear relaxation optimal solution is obtained. This methodology is called Branch and Price.

**Rules**

For a solution to the crew pairing problem to be considered feasible, the pairings must obey FAA regulations, union contract requirements and other airline specific rules. Such restrictive rules help reduce the size of the problem but can make the problem of determining an optimal crew schedule extremely complex. Below we describe some important rules governing legal pairings.

Within a duty period there are prescribed maximum and minimum rest times between flights. Typical values are 4 hours and 45 minutes respectively. The elapsed time in a duty period must be less than an allowable time limit, which is usually 12 hours. The total actual flying time in a duty must not exceed a limit, which is typically 8 hours. A duty conforming to these rules is said to be legal or valid. Similarly, a valid or legal pairing must satisfy some rules. Legal pairings may be composed of up to a maximum number of duty periods. A pairing must allow a minimum number of hours of rest between duties. This minimum rest period may need to be extended when the flying time in a twenty-four hour period exceeds eight hours [15].

## 2.2 Crew Rostering

Once the pairings have been constructed, the following problem to be solved is the assignment of the crew members to the pairings. This problem is known as the Crew Assignment Problem or the Crew Rostering Problem. There are many different crew rostering strategies used by the airlines. In North America, bid-line systems are normally preferred, in which the system creates anonymous schedules which then are assigned to the crew members through a bidding system that favours the most senior. In Europe a more personalized rostering approach is followed which takes into consideration crew preferences, availability, and skills.

The cost of crew rostering compared to the crew pairing problem is generally small. Thus, most formulated crew rostering problems aim to create a good quality of rosters in order to minimize cost. In bid-line models, the regularity of a schedule is mostly used to measure the roster’s quality [16]. When the rostering is assigned by the personalized assignment method, crew satisfaction is the target to maximize based on individual preferences [17]. In addition, there are formulations for both cases which aim to minimize the work time variation between schedules [18] [19].

The Crew schedules should follow many constraints imposed by regulation schemes and collective agreements. Examples of these constraints can be the limit of the consecutive or total working hours of a specific period, or the days off for the same period [8]. Other constraints to be used may be based on the experience of the crew members per pairing [20]. Additionally, constraints take into consideration qualification requirements or a minimum global satisfaction for all rosters [21]. Many constraints depending on the nature may be used as hard constraints or soft constraints with a penalty in the objective function. Some rules that govern the legality of a roster may be found in Figure 2.1.

Depending on the formulation of the crew assignment problem and the desired objective, many solution methodologies have been proposed. Heuristics or metaheuristics which work better for smaller problems have been developed by [19] and [18]. For larger problems, column generation methodologies seem to perform faster [22] [20].

| Type | Example |
| --- | --- |
| Rule | Less than 80 hrs flight time during the month |
| Cost | Quadratic penalty for deviation from 60 hrs flight time |
| Rule | Maximum six days of consecutive work |
| Cost | Large penalty for not assigning a paring |
| Rule | Minimum rest time |
| Cost | Penalty for short rest time |
| Cost | Penalty for early start after days off |
| Cost | Penalty for the assignment of a pairing that creates a conflict with time-off bids |

*Figure 2.1 Examples of horizontal rules and costs [23].*

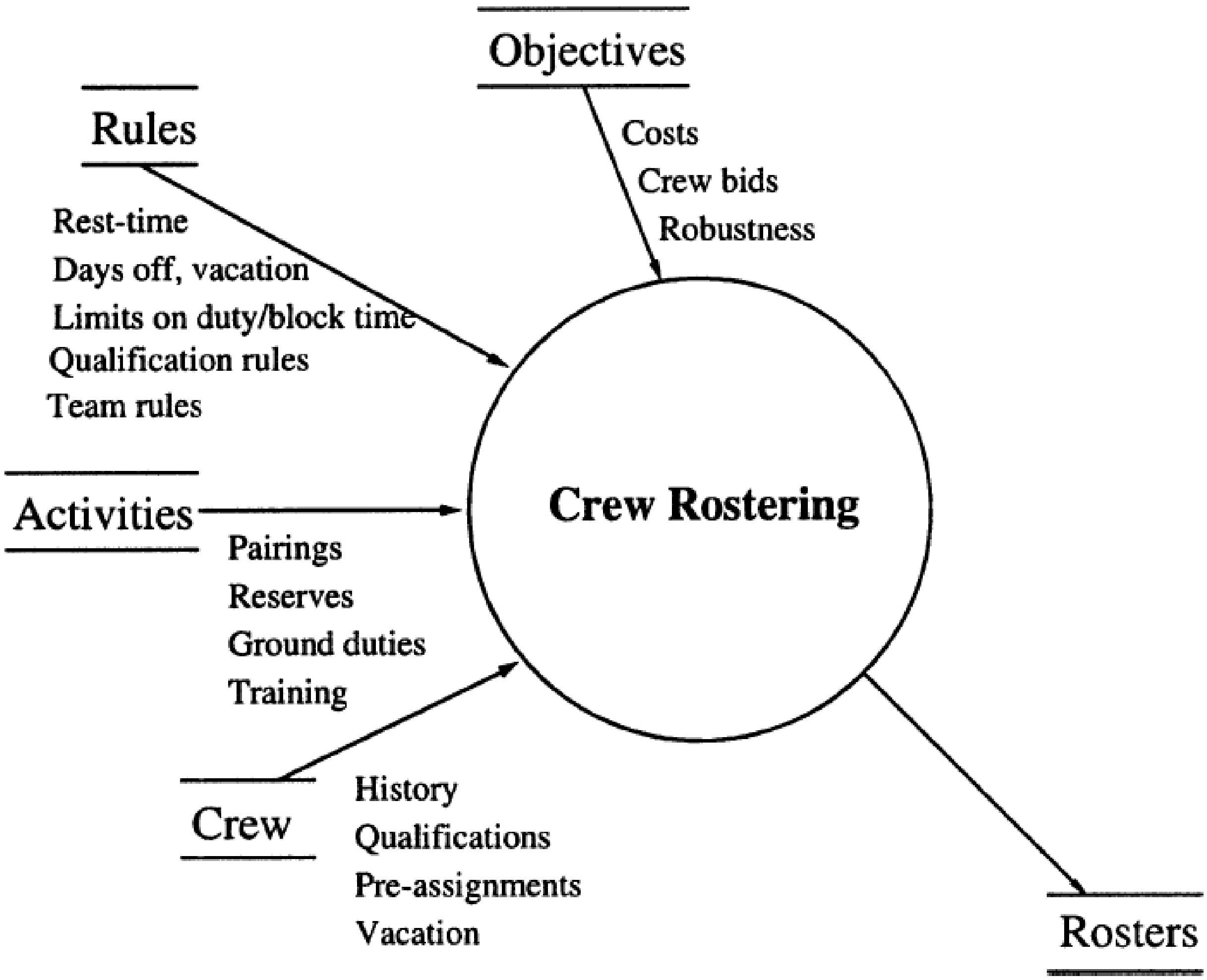


*Figure 2.2 Crew Rostering Problem [8].*

## 2.3 Crew Rostering with Min-Max Constraints

A type of regulation involves the constitution of many roster lines and crew members. These rules mainly govern the composition of the crew assigned to a pairing. Each pairing must be staffed correctly by the correct number of cabin personnel (e.g., captain, first officer) and service personnel (e.g. steward, hostess). The composition of the crew assigned to a pairing must comply with certain requirements concerning the qualifications of individual members and the number of crew members performing each function on board of the plane, e.g., the number of experienced crew members, language qualification, and crew members who must/cannot fly together [10]. The following lists some important examples of qualification-type constraints and state on which level they are defined [8].

- *Inexperienced crew members*. The number of inexperienced crewmembers on a particular pairing is usually bounded. In cockpit crew problems, at most one inexperienced pilot can operate a pairing. This constraint is usually found on the task level.
- *Must fly together*. In many cases, some crewmembers must be assigned the same activities, where the set of activities is not predefined. This rule could apply for instance to married couples for a certain number of pairings per month.
- *Incompatibilities*. Often, some crewmembers are incompatible in the sense that they cannot be assigned to the same pairing – and possibly not to some ground duties as well – due to conflicting personalities. This is primarily a cockpit problem and it occurs most frequently in long haul where the crew has to cooperate for a quite long period of time. This constraint is typically an example of a task-qualification.
- *Language qualification*. In the following, we give an important example of a typical leg-qualification constraint. However, note that most of the task-qualifications listed above could appear on a strict subset of the legs of the pairing and would then be leg-qualification constraints. In long-haul cabin crew problems these constraints are usually present. For example, on a Copenhagen–Bangkok flight leg we may need five Thai speaking cabin attendants. This is clearly a leg qualification. If all 15 cabin attendants on the flight fly the same pairing we could apply the qualification constraint on the pairing instead of the leg. Since the number of tasks is generally much smaller than the number of legs this will cause fewer constraints in the master problem. Unfortunately, the 15 cabin attendants may not fly the same pairing. Assume that 12 continue on the plane from Bangkok to Singapore (this leg only requires 12 cabin attendants) whereas 3 cabin attendants have a layover in Bangkok before they return to Copenhagen. Now the question is, how to “divide” the 5 language requirements between the 12-person pairing and the 3-person pairing? Since this question cannot be answered a priori, it must become a part of the optimization in the master problem and hence be modelled on the leg level.

These constraints can typically be treated as hard constraints whenever their violation may impair the security of the flight (e.g., crew qualifications, national legislation concerning duration of work and rest times) or as soft constraints preserving the social quality of the schedule (e.g., internal company rules, declared assignment preferences by the crew staff). In most airline companies the qualitative indicators have been integrated into the collective agreement and as such, they have been considered as (soft) constraints in the model [20] [10].

Additional constraints have also been used in a public transport assignment problem. These constraints are set to force a quality and legal schedule to a set cover problem [24]. But most of the constraints are set to tackle the language requirements, such research may be seen in [8] [21] [25] [10].

# 3 Model Formulation and Solution Methodology

In this chapter the mathematical model and its variations of the crew rostering problem will be thoroughly explained. The mathematical models, known as Set Cover and Set Partition, will be formulated with the aim to efficiently try to cover the maximum possible number of pairings with a specific set of crew member and explained. Adjustments to those models will be made in order to facilitate the need to have a maximum limit of crew members on a pairing with a specific characteristic, this will be succeeded with introducing the Min-Max constraints.

Later in this chapter, a branch and price methodology will be explained and fit to the forementioned problems. The column generation algorithm, used in other research [26] will be adapted to fit our case and produce only the most promising rosters for the problem. The column generation will be modelled as a shortest path problem with the side constraints to fulfil the rules from the authorities and the worker associations.

## 3.1 Master Problem formulation

In this section, the exact formulation of the set partition and the set cover master problem will be explained in further detail. Then, the adjustment made to both of them will be explained. These two problems with their two variations, adjusted and not, will be the Master Problems for the branch and price methodology.

### 3.1.1 Set Partition

For the crew rostering problem, it is needed to assign crew members to pairings in order to operate the sequence of flights. In this formulation of the problem, the crew members are assigned to the pairings. In addition, the pairings need a specific number of crew members which is normally called complement. The objective function aims to minimize the cost resulting from the assignment of the pairings to the crew members, based on the factors of interest.

**Sets:**

- $\boldsymbol{I}$: set of crew members,
- $\boldsymbol{R_i}$ : set of rosters of crew member $i$,
- $\boldsymbol{F}$ : set of flight routes.

**Parameters:**

- $\boldsymbol{c_{ij}}$ : cost of roster $j$ of crew member $i$,
- $\boldsymbol{h}$ : cost for each flight route that remains uncovered,
- $\boldsymbol{b_f}$ : complement (number of crew members required) of flight route $f$,
- $\boldsymbol{a_{ijf}}$ : binary parameter that takes the value 1 if roster $j$ of crew member $i$ covers flight route $f$, and 0 otherwise, $i \in I$, $j \in R_i$, $f \in F$.

**Decision Variables:**

- $\boldsymbol{x_{ij}}$ : binary decision variable that takes the value 1 if roster $j$ of crew member $i$ is assigned to this crew member, and 0 otherwise, $i \in I$, $j \in R_i$,
- $\boldsymbol{y_f}$ : binary decision variable that takes the value 1 if flight route $f$ remains uncovered, and 0 otherwise, $f \in F$.

Utilizing the above notation, the master problem is formulated as follows:

$$\text{Min} \sum_{i \in I} \sum_{j \in R_i} c_{ij} x_{ij} + \sum_{f \in F} h y_f \tag{1}$$

$$\text{s.t.} \sum_{j \in R_i} x_{ij} = 1, \ \forall i \in I \tag{2}$$

$$y_f + \sum_{i \in I} \sum_{j \in R_i} a_{ijf} x_{ij} = b_f, \ \forall f \in F \tag{3}$$

$$x_{ij} \text{ binary}, \ y_f \text{ integer}, \ \forall i, j, f \tag{4}$$

*Figure 3.1 Set Partition Formulation.*

The objective function (1) minimizes the total cost of the set partition formulation. This cost is the sum of the cost of the uncovered pairings plus the roster quality cost. Specific details on these costs are provided in the following paragraphs. Constraint set (2) ensures that exactly one roster, which as stated is a set of pairings for a specific crew member, is assigned to each crew member. Constraint set (3) states that each pairing must be covered by as many crew members as its complement dictates. Variable $y_f$ is used to denote the uncovered positions in flight route $f$; for each uncovered position, the corresponding penalty $h$ is imposed in the objective. Finally, constraint set (4) restricts the decision variables of the problem to integral values.

### Roster Quality cost

In the present thesis, we calculate the quality cost of each roster based on its deviation from the desired flight time window of the corresponding crew member. More specifically, each crew member has a desired flight time window $(LB_i,\ UB_i)$ for the total flight time to be assigned to him/her. The total flight time of each roster is equal to the sum of the flight time of all pairing assigned to it. If this flight time belongs to the flight time window $(LB_i,\ UB_i)$, the roster quality cost is zero. For any deviation from that window, the cost increases by a suitable factor, which for the present research will be set equal to 1.000. This is clearly depicted in the formulation below.

$$Penalized\ flight\ time\ of\ a\ roster = \begin{cases} Roster\ total\ flight\ time > UB_i \rightarrow\ Roster\ total\ flight\ time - UB_i \\ LB_i \leq Roster\ total\ flight\ time \leq UB_i \rightarrow 0 \\ Roster\ total\ flight\ time < LB_i \rightarrow\ UB_i - Roster\ total\ flight\ time \end{cases}$$

For example, a crew member with $(LB_i,\ UB_i) = (70, 78)$ and a roster with total flight time equal to 85 will incur a penalty cost $= (85 - 78) * 1000 = 7.000$.

The reason behind this cost and its formulation is valid. The extra hour for the crew members might bring discontent due to a bad work life balance and the airline will also need to pay the crew members in overtime costs. On the other hand, the low total flight time means the crew member is being underutilized and the extra time remaining available at the end of time period is still paid without providing any value.

### Cost of Uncovered Pairing

On the other hand, the cost for an uncovered pairing should be much greater that the cost from the roster quality cost. A pairing is a sequence of flight legs starting from one base airport and returning to the same base station. An uncovered pairing means that crew members from nearby home bases need to be outsourced for covering it. Thus, the covering of the pairings is understandably something the airlines prioritize, even over bad roster quality. This justifies the assigned cost for an uncovered pairing at 1.000.000 for the work presented in this thesis.

While uncovered pairings is something needed to be avoided, it is possible for some cases in a set of a problem to occur. These cases might have to do with regulations which won't allow crew members to operate pairings or by having multiple pairings at the same time which makes it impossible to cover with the existing number of crew members.

### 3.1.2 Set Cover

The set cover problem is a formulation which aims to cover as many pairings as possible without taking into consideration roster quality. This is a simplification of the set partition problem and the use of it is to see if all the pairings can be covered by the current set of crew members, which, as already stated is the previous section, is much more important than roster quality.

Sets:

$\boldsymbol{I}$: set of crew members,

$\boldsymbol{R_i}$: set of rosters of crew member $i$,

$\boldsymbol{F}$: set of flight routes.

Parameters:

$\boldsymbol{h}$ : cost for each flight route that remains uncovered,

$\boldsymbol{b_f}$ : complement (number of crew members required) of flight route $f$,

$\boldsymbol{a_{ijf}}$ : binary parameter that takes the value 1 if roster $j$ of crew member $i$ covers flight route $f$, and 0 otherwise, $i \in I$, $j \in R_i$, $f \in F$.

Decision Variables:

$\boldsymbol{x_{ij}}$ : binary decision variable that takes the value 1 if roster $j$ of crew member $i$ is assigned to this crew member, and 0 otherwise, $i \in I$, $j \in R_i$,

$\boldsymbol{y_f}$ : binary decision variable that takes the value 1 if flight route $f$ remains uncovered, and 0 otherwise, $f \in F$.

Utilizing the above notation, the master problem is formulated as follows:

$$\text{Min} \sum_{f \in F} h y_f \tag{5}$$

$$\text{s.t.} \sum_{j \in R_i} x_{ij} = 1, \ \forall i \in I \tag{6}$$

$$y_f + \sum_{i \in I} \sum_{j \in R_i} a_{ijf} x_{ij} \geq b_f, \ \forall f \in F \tag{7}$$

$$x_{ij} \text{ binary, } y_f \text{ integer, } \forall i, j, f \tag{8}$$

*Figure 3.2 Set Cover Formulation.*

The adjustments made to the set cover formulation is that in the objective functions (5) the cost regarding the quality of the rosters has been removed. The second adjustment is in constraint set (7) which monitors the covering of the pairings. Instead of the sum of the assigned crew members needed to be equal to the corresponding complement ($b_f$), we also allow it to be greater.

The optimal solution, although may include some pairings which are covered more times than the corresponding complement, will provide the best possible covering of the pairings. Besides utilizing the crew members optimally, the proposed approach exhibits an additional benefit, since the removal of the quality cost from the objective function and the relaxation of the constraint for the covering of the pairings reduces substantially the calculation time needed to reach the optimal solution. This change has a great positive impact on the performance of the column generation algorithm as will be demonstrated in the following chapters.

In addition, a mathematical model to transform the set cover solution to a corresponding set partition solution has been developed in [26]. This two-step approach, set cover first, followed by the set cover to set partition transformation, may not produce a better solution than the employment of the set partition from scratch, but it may provide an optimal solution faster. A major benefit may be realized if the solution given by the two-step approach is used as an initial solution for the quality phase of the column generation algorithm. This approach will not be part of this research.

### 3.1.3 Adjusted models with Minimum or Maximum Constraints

It many flights there may be additional restrictions that need to be respected. For example, specific flight legs may need pilots with specific experience such as landing in hard airports or there might be a special regulation to include a crew member who knows a specific language. To take into consideration such requirements, the below constraint is proposed to be added either to the set cover or the set partition problem:

$$\sum_{i\in I}\sum_{j\in R_i} p_i a_{ijf} x_{ij} \leq d_f, f \in F, \qquad (9)$$

with $p_i$ being a binary parameter which states if a particular crew member $i$ has a specific characteristic for which we want to impose a limit of $d_f$ on pairing $f$. This simple constraint added to the set partition or set cover problem formulates a problem that considers the desired criteria.

As seen above, this is an upper limit as the constraint is expressed as less or equal. This formulation works well for restrictions such as “not more that $d_f$ inexperienced crew members”, but cases which require at least a specific number of crew members with a desired characteristic are not directly covered. While changing the constraint to greater or equal is the intuitive thing to do, it is preferred for computational reasons for these cases to calculate the upper limit $d_f$ as the complements $b_f$ minus that requirement $r_f$ as seen below.

$$d_f = b_f - r_f \quad (10)$$

Although this adjustment will make these formulations able to provide a more desired solution, it is expected to increase the computational requirements needed for a solution. It is also expected to provide a higher objective function than the mathematical models without the additional constraints.

## 3.2 Branch and Price methodology

The models formulated in the previous sections need enormous computational resources in order to reach the optimal solution. To deal with this issue, a branch and price methodology is used. This is a very powerful methodology that combines Branch and Bound and Column Generation to address integer programs with a vast number of variables; thus, it has been applied to many scheduling problems even in the airline industry. The method has two key components, it efficiently adds legal columns (rosters) to the master problem that have the potential to improve the objective function and exploits an effective branching method which quickly narrows the solution space needed to be searched.

Beginning with the formulated model, the column generation adds a set of rosters that can potentially improve the objective function at each iteration. This is repeated until the column generation algorithm cannot find a roster that can improve the objective function. Then, the linear relaxation of the master problem is solved and branching takes place on one fractional variable. For each branching node, the column generation is employed again for identifying rosters that can potentially improve the linear relaxation solution. Each time branching takes place, a roster is assigned to a crew member. This procedure is repeated until an integral solution is obtained in which all crew members have been assigned a roster.

### 3.2.1 Column Generation

The column generation algorithm’s purpose is to construct rosters of pairings for a specific crew member which can improve the solution. Such rosters with negative reduced costs are calculated based on the optimal LP relaxation of the master problem. A negative reduced cost is an indication that the associated roster can improve the master objective; therefore, such rosters are added to the problem causing an update of the dual solution. Rosters that do not have negative reduced cost are not added to the master problem as they cannot improve its LP optimal solution. The column generation problem is solved as a shortest path network problem with side constrains, with the objective representing the corresponding reduced cost.

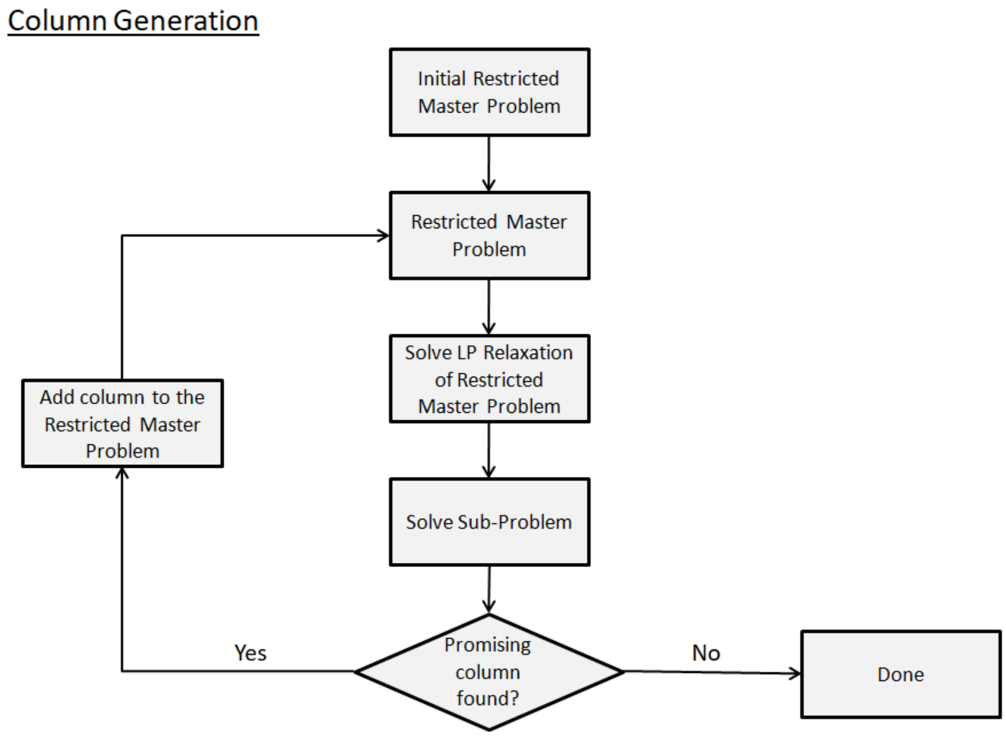


*Figure 3.3 Column Generation flowchart [26].*

**Modelling the shortest path network**

For the shortest path to traverse a network, the model $G = V, A$ is proposed. Each network path in $G$ is a roster, the network has a set of vertices $V$ which begin with a source node representing the crew member $(CM_i)$, a node for each pairing $(P_f)$, and finally a sink node (Fict.). The last node is a fictitious node representing the end of the path, a sink where all the paths terminate. The set of arches $A$ connect the nodes and are directional. An example is presented in Figure 3.4. In the example, the path CM1-P1, P1-P2, P2-Fict, represents a roster for crew member 1 with pairings 1 and 2.

The proposed network is a directed acyclic network. This means that there is no possible way to return to a node already traversed, this makes the total number of paths a finite number and the largest number of nodes traversed in any solution provided by the shortest path problem will not be larger than the total number of pairings plus two, the source and the sink nodes. The directed formulation allows the arches to be traversed only in the desired direction, so for example the paths CM1-P1 or CM1-P2 may be traversed but not the opposite.

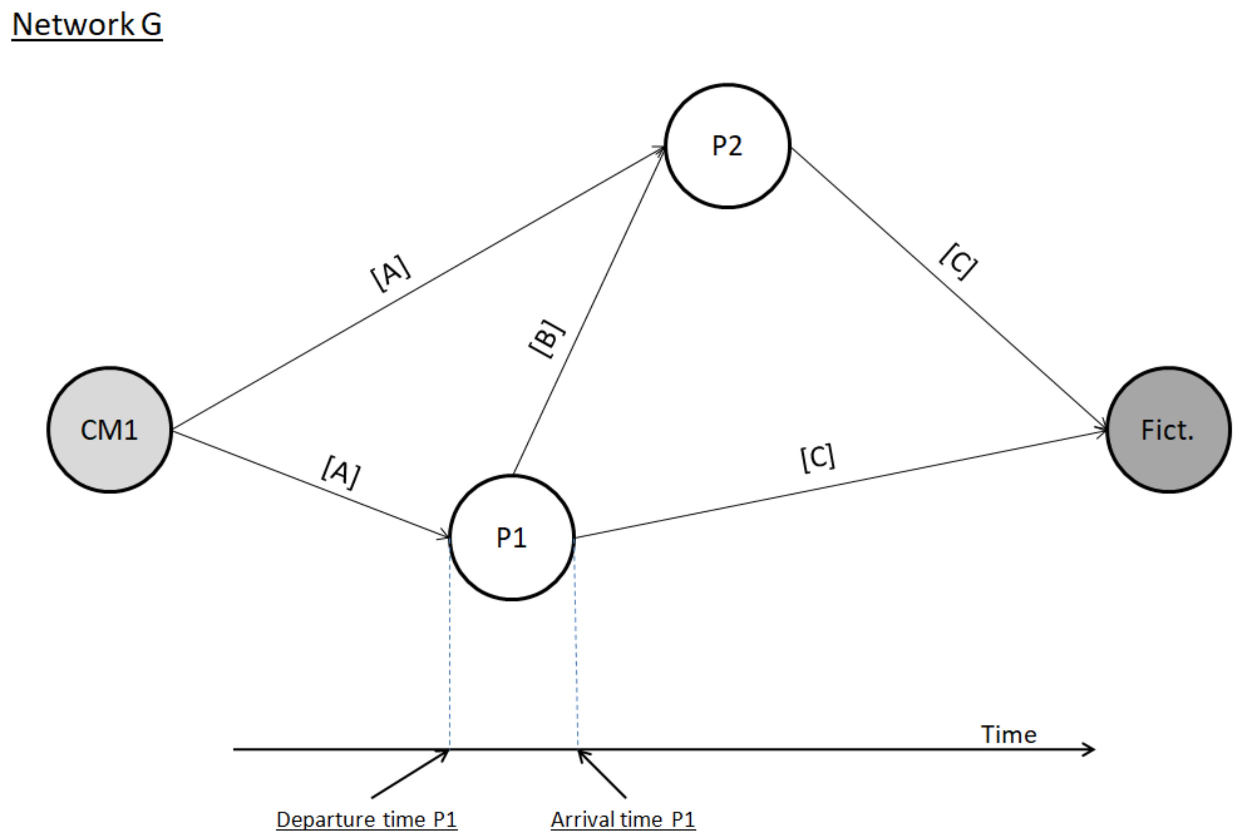


*Figure 3.4 Network G [26].*

This network is designed so that pairings are arranged in chronological order based on their departure time. Thus, a direct connection may exist only if the departure time of the next pairing is later than the arrival of the previous one. In our first example, the departure of pairing 2 was later that the arrival of pairing one, so an arch connecting the two nodes existed. In the second example seen in figure below, the pairing flight schedule overlaps so a path should not exist between them.

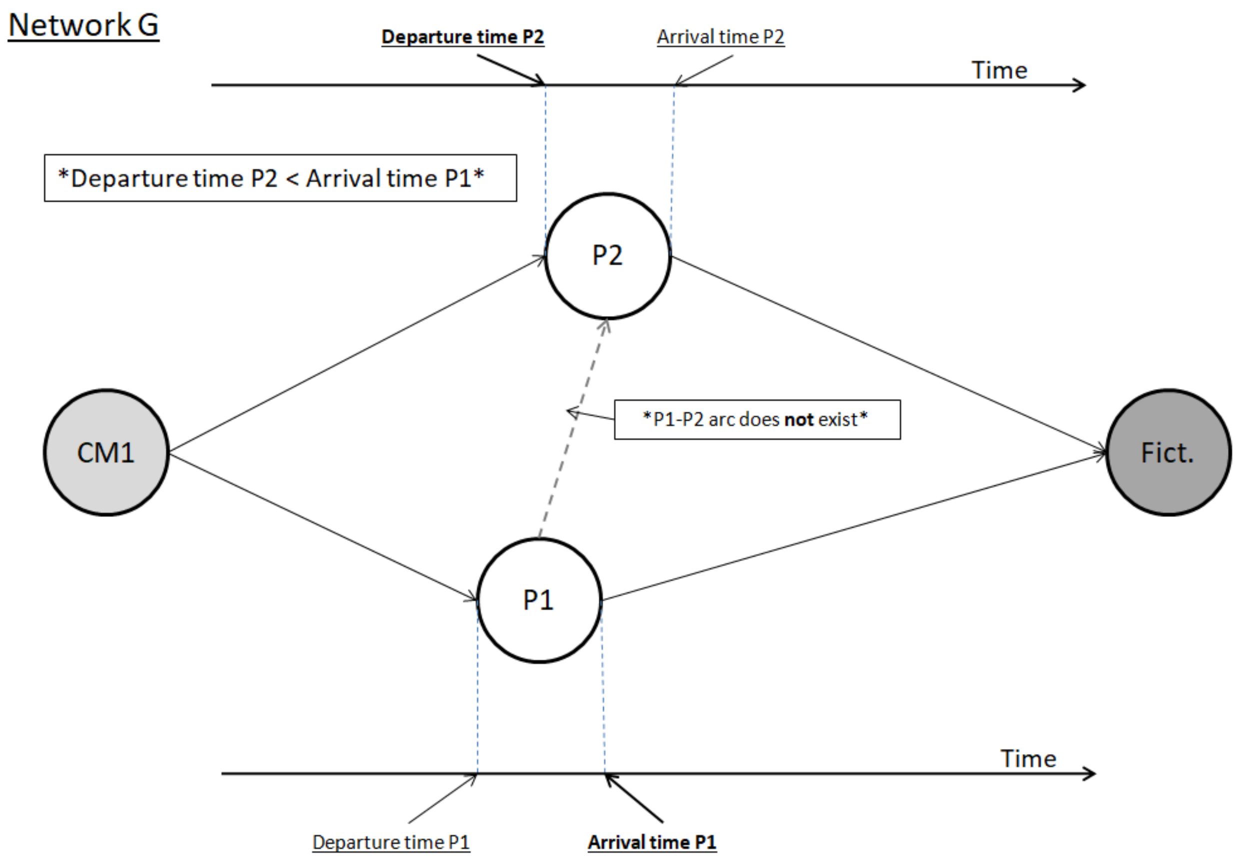


*Figure 3.5 An illegal connection between the nodes [26].*

### A Shortest Path Problem with Side Constraints

A shortest path problem with side constraints is a variant of the classical shortest path problem in which, in addition to finding the minimum length path between two nodes in a graph, we also need to satisfy some additional constraints. These constraints are referred to as "side constraints" because they are additional to the main objective of finding the shortest path.

For example, the constraints may be related to the maximum number of edges that can be traversed, the maximum time or cost that can be spent, or the presence of forbidden nodes or edges that must not be visited. The additional constraints make the problem more challenging and can lead to multiple solutions, some of which may not be feasible.

It's important to note that the addition of side constraints to a shortest path problem can significantly increase its complexity and render the identification of the optimal solution much more difficult. However, these problems are widely encountered in real-world applications, such as transportation planning, logistics, and network design, among others.

In the scope of the present thesis, the shortest path aims to maximize the negative reduced value, thus, the shortest path chosen from the algorithm will be a roster which will potentially reduce the objective function of the master problem. The side constraints will be the regulation scheme imposed by the authority or worker unions.

**Flight Regulation for crew members.**

The side constraints in this work are set to mimic the actual regulation scheme. The rules that need to be followed to determine the legality of a roster is included below.

1. *Maximum flight load*: The total flight time of a crew member is not permitted to overcome the Maximum Flight load for a specific planning horizon.
2. *Minimum Off Days*: The minimum days off a crew member should have in the specific planning horizon.
3. *XHoursBreak* / *YFlightHours*: This rule is to enforce the crew members rest at their home base for X hours between two pairings. The length X is determined by the Y hours of the arriving pairing.

For a day off to be valid, the time between two pairings should be over 34 hours, then for every 24 hours an extra day off is counted. In addition to the 34 hours minimum limit, there is a rule that states that for any flight arriving after 23:00 the next day cannot count as a day off. Similar for flight leaving before 06:00 the previous day cannot count as a day off.

For example, a crew member arriving from his pairing at 23:30 at day one and departing for his next pairing at 04:00 at day four, even when the time between the two pairings is over two days no day off is charged.

### 3.2.2 Branch and Bound

Once the column generation algorithm has finished and cannot find any other roster which may improve the objective function, the linear master problem has been solved. If the optimal solution of the linear master problem is integral, the search is completed and there is no need for further actions. Otherwise, two subproblems are created based on one variable, which although should be an integer, has a fractional value. For binary problems, such as the problems in the scope of this thesis, one subproblem or child problem will have this variable equal to 0 and the other equal to 1, as shown in the figure below. Then, the child problems are solved just as the parent problem and branching reoccurs if necessary. To efficiently move around the tree created by the branching actions, the tree node with the largest potential is selected and solved at each iteration. When an integer solution is found the search continues till the unsolved problems may not produce a potential better solution.

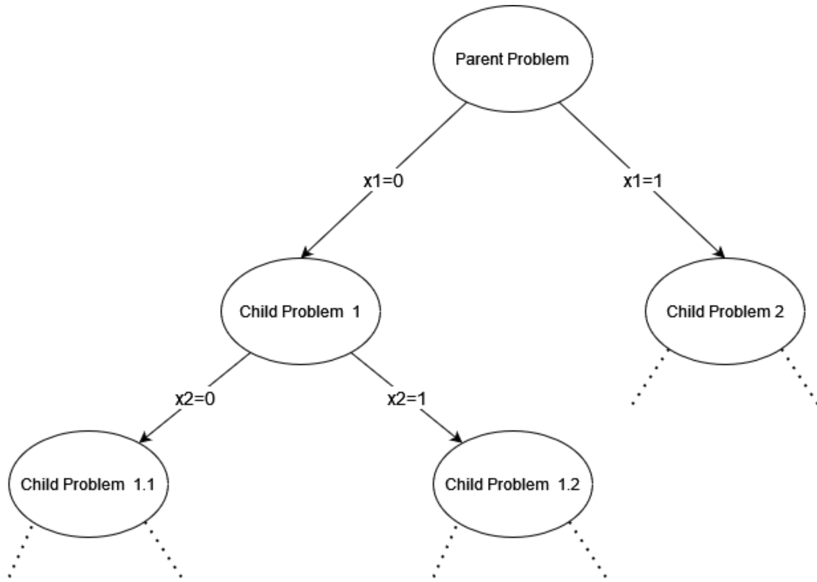


*Figure 3.6 Branching.*

In this work, variable selection for branching will be based on the largest fractional value of the roster variables. Then the right child problem, where the variable is set equal to one, is chosen to be solved again. The right child problem means an assignment of a crew member to a specific roster. All the other possible rosters for that particular crew member will be removed from the restricted master problem and the column generation algorithm will not consider him/her in the associated subtree for roster generation. If the pairing has been fully covered based on its complement, then all the other rosters containing this pairing which have not been assigned shall be removed from the restricted master problem. For the case of the left side child, only the branching variable is removed from the restricted master problem.

**Backtracking**

Backtracking occurs when the right side child cannot provide a feasible solution or the solution provided is much worse than the best linear relaxation of the problem. This is an indication that the candidate problem does not have much potential; thus, we choose to move to another candidate subproblem on which we have not yet applied column generation. The candidate problem which is selected is the one with the best bound on the optimal master objective.

**Termination Conditions**

Although the number of feasible solutions is finite, enormous computational resources are required for identifying all of them and selecting the optimal. For that reason, rules have been set up which decide if the search after finding an integral solution shall continue or not.

- *MIP Gap Tolerance:* To be sure that our solution is close to optimal, we check the absolute and relative MIP gap tolerance. If the absolute and the relative difference between the objective function of the integer solution and the objective function of the master problem in the root node are both greater than the MIP gaps defined, then we continue our search for a better integer solution.
- *Maximum Number of Backtracks:* After the first integral solution has been found, backtracking may only occur up to a specific limit.
- *Maximum Number of Tree Nodes created:* After the first integral solution has been found, the number of children problems created are counted and put to a limit.

If a limit is surpassed, the best solution so far is selected.

## 3.3 An overview of the methodology with an Example

### 3.3.1 Overview of Branch and Price methodology

As every part of the branch and price methodology has been described, in this section a complete overview is presented to understand it better. The flow of the branch and price methodology may be seen in Figure 3.7. The formulated problems described in the beginning of this chapter will be used as the master problem. For the beginning of the column generation process a restricted master problem is used, with an empty roster for each crew member and all pairings uncovered. Then, the column generation gradually adds the rosters which have a potential to improve the objective function based on their reduced costs. In the column generation process, only legal rosters are accepted. Once the column generation is finished and no other roster which may improve the objective function can be identified, branching takes place if the solution is not integral. The roster variable which is chosen for branching is the one with the highest fractional value. Then, two child problems are created by adjusting the restricted master problem. On the left side, child problem one is created by removing the fractional roster variable from the problem. On the right side, child problem two is created by removing all the other roster variables of the associated crew member and assigning the fractional roster variable to him/her. This implies that the right side branching process assigns a specific roster to this crew member. In the column generation process of the associated subtree, no other roster will be generated for this crew member and pairings whose complement has been fulfilled will also not be considered. As the right side branching moves closer to a solution, that child problem is the next problem on which column generation is applied. This process continues until an integral solution is found. Once the integral solution is found, backtracking may be applied if needed to find a better solution.

Branch and Price

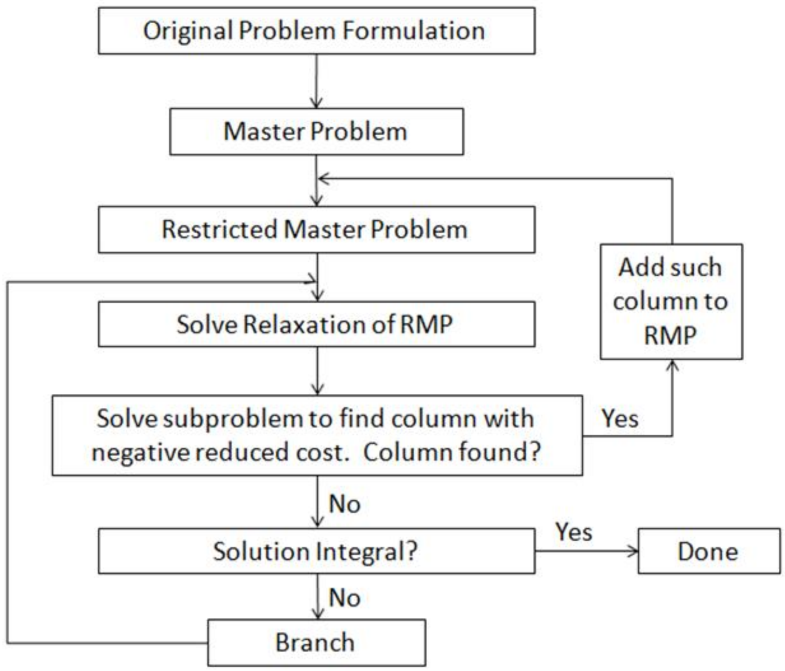


*Figure 3.7 Branch and Price flowchart [27].*

### 3.3.2 Example case study

In this example case study, the process is presented step by step. The problem aims for the assignment of $N = 4$ crew members to $R = 6$ pairings in a set partition formulation with Min-Max constraints. The set partition formulation with Min-Max constraints incorporates in full all the presented details. The data for the crew members may be seen in Table 3.1, where $LB_i$ is the lower bound and $UB_i$ the upper bound of the desired window of flight hours. The column $p_i$ is a Boolean parameter for the characteristic which will be in the Min-Max constraints. So only the crew members 0 and crew member 2 with $p_i = 1$ will participate in the Min-Max constraints.

*Table 3.1 Crew Members Data*

| $N$ | $LB_i$ | $UB_i$ | $p_i$ |
|---|---|---|---|
| **0** | 35 | 41 | 1 |
| **1** | 52 | 58 | 0 |
| **2** | 75 | 81 | 1 |
| **3** | 25 | 31 | 0 |

The data for the pairings are presented in Table 3.2. For each of the six flights we have the Day of Departure ($DD_f$), the Day of Arrival ($DA_f$), the Departure Time ($DT_f$), the Arrival Time ($AT_f$), Flying Hours ($FH_f$), the complements ($b_f$), and the limit of Min-Max constraints ($d_f$).

*Table 3.2 Pairing Data*

| $R$ | $DD_f$ | $DA_f$ | $DT_f$ | $AT_f$ | $FH_f$ | $b_f$ | $d_f$ |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 2 | 16:20 | 00:20 | 8 | 2 | 1 |
| **1** | 10 | 11 | 07:20 | 8:20 | 25 | 2 | 1 |
| **2** | 11 | 11 | 01:50 | 14:50 | 13 | 2 | 1 |
| **3** | 11 | 12 | 19:40 | 16:40 | 21 | 2 | 1 |
| **4** | 14 | 15 | 23:00 | 17:00 | 18 | 2 | 1 |
| **5** | 18 | 18 | 09:10 | 14:10 | 5 | 2 | 1 |

The restricted master problem created to initiate the branch and price methodology is shown in Figure 3.8. As shown, variables x1 to x6 are the empty positions for each pairing (or routes in the figure), in the objective function they have a cost of 1.000.000 which is the cost for an uncovered pairing. The variables x1 to x6 are only seen in the pairing constraints as they are not connected to any crew members. The variables x7 to x10 are the rosters for each crew member with no pairing assigned. The cost in the objective function is based on the difference from the target window times 1.000, which is the penalty of the deviation. For example, x7 which is the empty roster of crew member 0 has zero flight hours. The lower bound of crew member 0 is 35 so the cost in the objective function is $(35 - 0) * 1.000 = 35.000$. The Min-Max constraints do not have any variable as no pairing is assigned to a roster yet.

```
Minimize
 obj1: 1000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 1000000 x5
       + 1000000 x6 + 35000 x7 + 52000 x8 + 75000 x9 + 25000 x10
Subject To
 CrewMember0: x7  = 1
 CrewMember1: x8  = 1
 CrewMember2: x9  = 1
 CrewMember3: x10  = 1
 Route0:      x1  = 2
 Route1:      x2  = 2
 Route2:      x3  = 2
 Route3:      x4  = 2
 Route4:      x5  = 2
 Route5:      x6  = 2
 MinMax0:     <= 1
 MinMax1:     <= 1
 MinMax2:     <= 1
 MinMax3:     <= 1
 MinMax4:     <= 1
 MinMax5:     <= 1
Bounds
 0 <= x1 <= 2
 0 <= x2 <= 2
 0 <= x3 <= 2
 0 <= x4 <= 2
 0 <= x5 <= 2
 0 <= x6 <= 2
End
```

*Figure 3.8 Beginning Restricted Master Problem.*

In Figure 3.10, the column generation has added all promising rosters to the restricted master problem for crew member 0. For example, variable x16 has been added which consists of the pairings 0, 1, and 3. As crew member 0 has the characteristic, which is limited in the pairings, further than the pairing constraints, the variable is seen in the Min-Max constraints. The cost in the objective functions is calculated as $((FH_0 + FH_1 + FH_3) - UB_0) * 1000 = ((8 + 25 + 21) - 41) * 1.000 = 13.000$.

The rosters are added based on the negative reduced cost which is calculated based on the duals. Continuing with roster x16 the negative reduced costs are calculated by subtracting the corresponding duals, see in Figure 3.9, from the cost in the objective function. So, the negative reduced cost is $13.000 - (35.000 + 1.000.000 + 1.000.000 + 1.000.000) = -3.022.000$.

```
Display dual values for which constraint(s): *
Constraint Name               Dual Price
CrewMember0                 35000.000000
CrewMember1                 52000.000000
CrewMember2                 75000.000000
CrewMember3                 25000.000000
Route0                    1000000.000000
Route1                    1000000.000000
Route2                    1000000.000000
Route3                    1000000.000000
Route4                    1000000.000000
Route5                    1000000.000000
All other dual prices matching '*' are 0.
```

*Figure 3.9 Dual of Master restricted problem.*

```
Minimize
 obj1: 1000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 1000000 x5
       + 1000000 x6 + 35000 x7 + 52000 x8 + 75000 x9 + 25000 x10 + 27000 x11
       + 10000 x12 + 2000 x13 + 22000 x14 + 14000 x15 + 13000 x16 + 1000 x17
       + 1000 x18 + 19000 x19 + 31000 x21 + 24000 x22 + 36000 x23 + 3000 x24
Subject To
 CrewMember0: x7 + x11 + x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x20
              + x21 + x22 + x23 + x24  = 1
 CrewMember1: x8  = 1
 CrewMember2: x9  = 1
 CrewMember3: x10  = 1
 Route0:      x1 + x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23
              + x24  = 2
 Route1:      x2 + x12 + x13 + x16 + x21 + x23  = 2
 Route2:      x3 + x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24  = 2
 Route3:      x4 + x16 + x17 + x18 + x19 + x21 + x22 + x23  = 2
 Route4:      x5 + x19 + x20 + x21 + x22 + x23 + x24  = 2
 Route5:      x6 + x22 + x23 + x24  = 2
 MinMax0:     x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23 + x24
              <= 1
 MinMax1:     x12 + x13 + x16 + x21 + x23 <= 1
 MinMax2:     x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 <= 1
 MinMax3:     x16 + x17 + x18 + x19 + x21 + x22 + x23 <= 1
 MinMax4:     x19 + x20 + x21 + x22 + x23 + x24 <= 1
 MinMax5:     x22 + x23 + x24 <= 1
```

*Figure 3.10 Rosters added to RMP for Crew Member 0.*

In the second iteration of the column generation, the promising rosters of crew member 1 are added. Similarly to crew member 0, the rosters have negative reduced cost and the penalty in the objective function is calculated as normally. Here an example variable x38, which is a roster with pairings 0, 1, 3, 4, and 5, is seen in the pairing constraints but not the Min-Max constraints. This is because crew member does not have the characteristic which is being limited on the pairings.

```
Minimize
 obj1: 1000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 1000000 x5
       + 1000000 x6 + 35000 x7 + 52000 x8 + 75000 x9 + 25000 x10 + 27000 x11
       + 10000 x12 + 2000 x13 + 22000 x14 + 14000 x15 + 13000 x16 + 1000 x17
       + 1000 x18 + 19000 x19 + 31000 x21 + 24000 x22 + 36000 x23 + 3000 x24
       + 44000 x25 + 27000 x26 + 19000 x27 + 39000 x28 + 31000 x29 + 10000 x31
       + 6000 x32 + 1000 x33 + 2000 x34 + 14000 x35 + 7000 x37 + 19000 x38
Subject To
 CrewMember0: x7 + x11 + x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x20
              + x21 + x22 + x23 + x24  = 1
 CrewMember1: x8 + x25 + x26 + x27 + x28 + x29 + x30 + x31 + x32 + x33 + x34
              + x35 + x36 + x37 + x38  = 1
 CrewMember2: x9  = 1
 CrewMember3: x10  = 1
 Route0:      x1 + x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23
              + x24 + x25 + x27 + x29 + x30 + x31 + x33 + x34 + x35 + x36 + x37
              + x38  = 2
 Route1:      x2 + x12 + x13 + x16 + x21 + x23 + x26 + x27 + x30 + x32 + x33
              + x35 + x36 + x38  = 2
 Route2:      x3 + x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 + x28 + x29
              + x31 + x34 + x37  = 2
 Route3:      x4 + x16 + x17 + x18 + x19 + x21 + x22 + x23 + x30 + x31 + x32
              + x34 + x35 + x37 + x38  = 2
 Route4:      x5 + x19 + x20 + x21 + x22 + x23 + x24 + x33 + x34 + x35 + x36
              + x37 + x38  = 2
 Route5:      x6 + x22 + x23 + x24 + x36 + x37 + x38  = 2
 MinMax0:     x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23 + x24
              <= 1
 MinMax1:     x12 + x13 + x16 + x21 + x23 <= 1
 MinMax2:     x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 <= 1
 MinMax3:     x16 + x17 + x18 + x19 + x21 + x22 + x23 <= 1
 MinMax4:     x19 + x20 + x21 + x22 + x23 + x24 <= 1
 MinMax5:     x22 + x23 + x24 <= 1
```

*Figure 3.11 Rosters added to RMP for Crew Member 1.*

In Figure 3.12, the rosters have been added also for crew member 2 and crew member 3. Also here, it is seen that in the Min-Max constraints only rosters of crew member 2 are added.

```
Minimize
 obj1: 1000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 1000000 x5
       + 1000000 x6 + 35000 x7 + 52000 x8 + 75000 x9 + 25000 x10 + 27000 x11
       + 10000 x12 + 2000 x13 + 22000 x14 + 14000 x15 + 13000 x16 + 1000 x17
       + 1000 x18 + 19000 x19 + 31000 x21 + 24000 x22 + 36000 x23 + 3000 x24
       + 44000 x25 + 27000 x26 + 19000 x27 + 39000 x28 + 31000 x29 + 10000 x31
       + 6000 x32 + 1000 x33 + 2000 x34 + 14000 x35 + 7000 x37 + 19000 x38
       + 67000 x39 + 50000 x40 + 42000 x41 + 62000 x42 + 54000 x43 + 21000 x44
       + 33000 x45 + 29000 x46 + 11000 x47 + 15000 x48 + 3000 x49 + 6000 x50
       + 10000 x51 + 2000 x54 + 12000 x55 + 4000 x56 + 3000 x57 + 11000 x58
       + 15000 x59 + 21000 x61 + 12000 x62 + 5000 x63 + 8000 x64
Subject To
 CrewMember0: x7 + x11 + x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x20
              + x21 + x22 + x23 + x24  = 1
 CrewMember1: x8 + x25 + x26 + x27 + x28 + x29 + x30 + x31 + x32 + x33 + x34
              + x35 + x36 + x37 + x38  = 1
 CrewMember2: x9 + x39 + x40 + x41 + x42 + x43 + x44 + x45 + x46 + x47 + x48
              + x49 + x50 + x51 + x52  = 1
 CrewMember3: x10 + x53 + x54 + x55 + x56 + x57 + x58 + x59 + x60 + x61 + x62
              + x63 + x64 + x65  = 1
 Route0:      x1 + x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23
              + x24 + x25 + x27 + x29 + x30 + x31 + x33 + x34 + x35 + x36 + x37
              + x38 + x39 + x41 + x43 + x44 + x45 + x48 + x49 + x51 + x52 + x54
              + x56 + x58  = 2
 Route1:      x2 + x12 + x13 + x16 + x21 + x23 + x26 + x27 + x30 + x32 + x33
              + x35 + x36 + x38 + x40 + x41 + x44 + x46 + x47 + x49 + x50 + x52
              + x53 + x54 + x59 + x62 + x65  = 2
 Route2:      x3 + x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 + x28 + x29
              + x31 + x34 + x37 + x42 + x43 + x45 + x48 + x51 + x55 + x56 + x57
              + x58 + x60 + x61 + x63 + x64  = 2
 Route3:      x4 + x16 + x17 + x18 + x19 + x21 + x22 + x23 + x30 + x31 + x32
              + x34 + x35 + x37 + x38 + x44 + x45 + x46 + x47 + x48 + x49 + x50
              + x51 + x52 + x57 + x58 + x59 + x61 + x64  = 2
 Route4:      x5 + x19 + x20 + x21 + x22 + x23 + x24 + x33 + x34 + x35 + x36
              + x37 + x38 + x47 + x48 + x49 + x50 + x51 + x52 + x60 + x61 + x62
              + x63  = 2
 Route5:      x6 + x22 + x23 + x24 + x36 + x37 + x38 + x50 + x51 + x52 + x63
              + x64 + x65  = 2
 MinMax0:     x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23 + x24
              + x39 + x41 + x43 + x44 + x45 + x48 + x49 + x51 + x52 <= 1
 MinMax1:     x12 + x13 + x16 + x21 + x23 + x40 + x41 + x44 + x46 + x47 + x49
              + x50 + x52 <= 1
 MinMax2:     x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 + x42 + x43 + x45
              + x48 + x51 <= 1
 MinMax3:     x16 + x17 + x18 + x19 + x21 + x22 + x23 + x44 + x45 + x46 + x47
              + x48 + x49 + x50 + x51 + x52 <= 1
 MinMax4:     x19 + x20 + x21 + x22 + x23 + x24 + x47 + x48 + x49 + x50 + x51
              + x52 <= 1
 MinMax5:     x22 + x23 + x24 + x50 + x51 + x52 <= 1
```

*Figure 3.12 Rosters added to RMP for Crew Member 2 & 3.*

Once finished with one iteration for each crew member, the column generation repeats again the calculations for crew member 0 as the restricted master problem has changed since last time. Now, three different promising rosters are added for crew member 0.

```
Minimize
 obj1: 1000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 1000000 x5
       + 1000000 x6 + 35000 x7 + 52000 x8 + 75000 x9 + 25000 x10 + 27000 x11
       + 10000 x12 + 2000 x13 + 22000 x14 + 14000 x15 + 13000 x16 + 1000 x17
       + 1000 x18 + 19000 x19 + 31000 x21 + 24000 x22 + 36000 x23 + 3000 x24
       + 44000 x25 + 27000 x26 + 19000 x27 + 39000 x28 + 31000 x29 + 10000 x31
       + 6000 x32 + 1000 x33 + 2000 x34 + 14000 x35 + 7000 x37 + 19000 x38
       + 67000 x39 + 50000 x40 + 42000 x41 + 62000 x42 + 54000 x43 + 21000 x44
       + 33000 x45 + 29000 x46 + 11000 x47 + 15000 x48 + 3000 x49 + 6000 x50
       + 10000 x51 + 2000 x54 + 12000 x55 + 4000 x56 + 3000 x57 + 11000 x58
       + 15000 x59 + 21000 x61 + 12000 x62 + 5000 x63 + 8000 x64 + 17000 x66
       + 9000 x67 + 5000 x68
Subject To
 CrewMember0: x7 + x11 + x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x20
              + x21 + x22 + x23 + x24 + x66 + x67 + x68  = 1
 CrewMember1: x8 + x25 + x26 + x27 + x28 + x29 + x30 + x31 + x32 + x33 + x34
              + x35 + x36 + x37 + x38  = 1
 CrewMember2: x9 + x39 + x40 + x41 + x42 + x43 + x44 + x45 + x46 + x47 + x48
              + x49 + x50 + x51 + x52  = 1
 CrewMember3: x10 + x53 + x54 + x55 + x56 + x57 + x58 + x59 + x60 + x61 + x62
              + x63 + x64 + x65  = 1
 Route0:      x1 + x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23
              + x24 + x25 + x27 + x29 + x30 + x31 + x33 + x34 + x35 + x36 + x37
              + x38 + x39 + x41 + x43 + x44 + x45 + x48 + x49 + x51 + x52 + x54
              + x56 + x58 + x67  = 2
 Route1:      x2 + x12 + x13 + x16 + x21 + x23 + x26 + x27 + x30 + x32 + x33
              + x35 + x36 + x38 + x40 + x41 + x44 + x46 + x47 + x49 + x50 + x52
              + x53 + x54 + x59 + x62 + x65 + x68  = 2
 Route2:      x3 + x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 + x28 + x29
              + x31 + x34 + x37 + x42 + x43 + x45 + x48 + x51 + x55 + x56 + x57
              + x58 + x60 + x61 + x63 + x64 + x66 + x67  = 2
 Route3:      x4 + x16 + x17 + x18 + x19 + x21 + x22 + x23 + x30 + x31 + x32
              + x34 + x35 + x37 + x38 + x44 + x45 + x46 + x47 + x48 + x49 + x50
              + x51 + x52 + x57 + x58 + x59 + x61 + x64  = 2
 Route4:      x5 + x19 + x20 + x21 + x22 + x23 + x24 + x33 + x34 + x35 + x36
              + x37 + x38 + x47 + x48 + x49 + x50 + x51 + x52 + x60 + x61 + x62
              + x63  = 2
 Route5:      x6 + x22 + x23 + x24 + x36 + x37 + x38 + x50 + x51 + x52 + x63
              + x64 + x65 + x66 + x67 + x68  = 2
 MinMax0:     x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23 + x24
              + x39 + x41 + x43 + x44 + x45 + x48 + x49 + x51 + x52 + x67 <= 1
 MinMax1:     x12 + x13 + x16 + x21 + x23 + x40 + x41 + x44 + x46 + x47 + x49
              + x50 + x52 + x68 <= 1
 MinMax2:     x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 + x42 + x43 + x45
              + x48 + x51 + x66 + x67 <= 1
 MinMax3:     x16 + x17 + x18 + x19 + x21 + x22 + x23 + x44 + x45 + x46 + x47
              + x48 + x49 + x50 + x51 + x52 <= 1
 MinMax4:     x19 + x20 + x21 + x22 + x23 + x24 + x47 + x48 + x49 + x50 + x51
              + x52 <= 1
 MinMax5:     x22 + x23 + x24 + x50 + x51 + x52 + x66 + x67 + x68 <= 1
```

*Figure 3.13 Rosters added to RMP for Crew Member 0 in second iteration.*

The column generation process continues adding promising rosters to the restricted master problem until no other can be identified. In Figure 3.14 we see all the rosters created for the restricted master problem. In total 61 promising rosters have been added to the problem. Before the next step, which is the branch and bound, it is good to point out a few maybe details which might be overseen. For example, no roster has both pairing 1 and pairing 2, as those pairings for some time overlap. In addition, some variables such as x65 do not appear in the objective function, this is due to the fact that its total flight time lies within the desired time window, thus the cost being equal to zero.

```
Minimize
 obj1: 1000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 1000000 x5
       + 1000000 x6 + 35000 x7 + 52000 x8 + 75000 x9 + 25000 x10 + 27000 x11
       + 10000 x12 + 2000 x13 + 22000 x14 + 14000 x15 + 13000 x16 + 1000 x17
       + 1000 x18 + 19000 x19 + 31000 x21 + 24000 x22 + 36000 x23 + 3000 x24
       + 44000 x25 + 27000 x26 + 19000 x27 + 39000 x28 + 31000 x29 + 10000 x31
       + 6000 x32 + 1000 x33 + 2000 x34 + 14000 x35 + 7000 x37 + 19000 x38
       + 67000 x39 + 50000 x40 + 42000 x41 + 62000 x42 + 54000 x43 + 21000 x44
       + 33000 x45 + 29000 x46 + 11000 x47 + 15000 x48 + 3000 x49 + 6000 x50
       + 10000 x51 + 2000 x54 + 12000 x55 + 4000 x56 + 3000 x57 + 11000 x58
       + 15000 x59 + 21000 x61 + 12000 x62 + 5000 x63 + 8000 x64 + 17000 x66
       + 9000 x67 + 5000 x68 + 6000 x69 + 1000 x70 + 1000 x71
Subject To
 CrewMember0: x7 + x11 + x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x20
              + x21 + x22 + x23 + x24 + x66 + x67 + x68  = 1
 CrewMember1: x8 + x25 + x26 + x27 + x28 + x29 + x30 + x31 + x32 + x33 + x34
              + x35 + x36 + x37 + x38 + x69 + x70 + x71  = 1
 CrewMember2: x9 + x39 + x40 + x41 + x42 + x43 + x44 + x45 + x46 + x47 + x48
              + x49 + x50 + x51 + x52  = 1
 CrewMember3: x10 + x53 + x54 + x55 + x56 + x57 + x58 + x59 + x60 + x61 + x62
              + x63 + x64 + x65  = 1
 Route0:      x1 + x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23
              + x24 + x25 + x27 + x29 + x30 + x31 + x33 + x34 + x35 + x36 + x37
              + x38 + x39 + x41 + x43 + x44 + x45 + x48 + x49 + x51 + x52 + x54
              + x56 + x58 + x67 + x71  = 2
 Route1:      x2 + x12 + x13 + x16 + x21 + x23 + x26 + x27 + x30 + x32 + x33
              + x35 + x36 + x38 + x40 + x41 + x44 + x46 + x47 + x49 + x50 + x52
              + x53 + x54 + x59 + x62 + x65 + x68 + x69 + x70 + x71  = 2
 Route2:      x3 + x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 + x28 + x29
              + x31 + x34 + x37 + x42 + x43 + x45 + x48 + x51 + x55 + x56 + x57
              + x58 + x60 + x61 + x63 + x64 + x66 + x67  = 2
 Route3:      x4 + x16 + x17 + x18 + x19 + x21 + x22 + x23 + x30 + x31 + x32
              + x34 + x35 + x37 + x38 + x44 + x45 + x46 + x47 + x48 + x49 + x50
              + x51 + x52 + x57 + x58 + x59 + x61 + x64 + x69 + x70 + x71  = 2
 Route4:      x5 + x19 + x20 + x21 + x22 + x23 + x24 + x33 + x34 + x35 + x36
              + x37 + x38 + x47 + x48 + x49 + x50 + x51 + x52 + x60 + x61 + x62
              + x63 + x69  = 2
 Route5:      x6 + x22 + x23 + x24 + x36 + x37 + x38 + x50 + x51 + x52 + x63
              + x64 + x65 + x66 + x67 + x68 + x70 + x71  = 2
 MinMax0:     x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23 + x24
              + x39 + x41 + x43 + x44 + x45 + x48 + x49 + x51 + x52 + x67 <= 1
 MinMax1:     x12 + x13 + x16 + x21 + x23 + x40 + x41 + x44 + x46 + x47 + x49
              + x50 + x52 + x68 <= 1
 MinMax2:     x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 + x42 + x43 + x45
              + x48 + x51 + x66 + x67 <= 1
 MinMax3:     x16 + x17 + x18 + x19 + x21 + x22 + x23 + x44 + x45 + x46 + x47
              + x48 + x49 + x50 + x51 + x52 <= 1
 MinMax4:     x19 + x20 + x21 + x22 + x23 + x24 + x47 + x48 + x49 + x50 + x51
              + x52 <= 1
 MinMax5:     x22 + x23 + x24 + x50 + x51 + x52 + x66 + x67 + x68 <= 1
```

*Figure 3.14 All rosters generated by the Column Generation algorithm.*

For the branch and bound, the chosen variable to branch on is x50 as that variable has the greatest fractional value. From branching, two children problems are created, the left side or child problem one where x50 is equal to 0, so the roster is removed from the problem, and the right side or child problem two where x50 = 1, so this roster is assigned to the crew member.

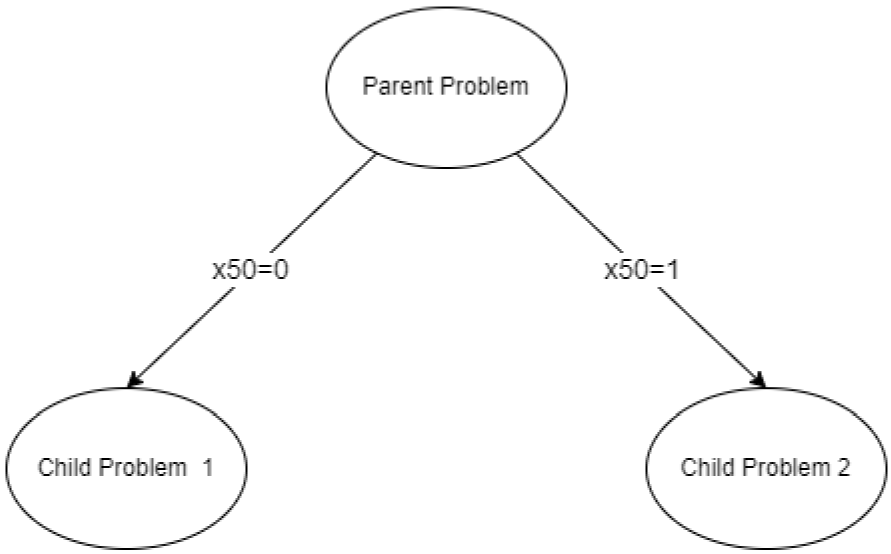


*Figure 3.15 Branching on variable x50.*

At the right side as seen in Figure 3.16, by assigning the roster to the crew member we can remove all the other rosters of this crew member from the master problem. Thus, for crew member 2, the only roster which will remain will be the one on which branching occurred, now x25 as other variables have been removed. This will also happen to the pairing constraints and the Min-Max constraints if the complement or Min-Max limit has been achieved. As the crew member has the characteristic and in the Min-Max constrains there is a limit of one, all the rosters containing the pairings of the branching variable will be removed. The branching roster includes the pairings 0, 2, 3, 4, and 5, so the rosters of crew member 0 which have those pairings are removed.

```
Minimize
 obj1: 1000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 1000000 x5
       + 1000000 x6 + 35000 x7 + 52000 x8 + 25000 x9 + 10000 x10 + 44000 x11
       + 27000 x12 + 19000 x13 + 39000 x14 + 31000 x15 + 10000 x17 + 6000 x18
       + 1000 x19 + 2000 x20 + 14000 x21 + 7000 x23 + 19000 x24 + 10000 x25
       + 2000 x27 + 12000 x28 + 4000 x29 + 3000 x30 + 11000 x31 + 15000 x32
       + 21000 x34 + 12000 x35 + 5000 x36 + 8000 x37 + 6000 x39 + 1000 x40
       + 1000 x41
Subject To
 CrewMember0: x7 + x10  = 1
 CrewMember1: x8 + x11 + x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x20
              + x21 + x22 + x23 + x24 + x39 + x40 + x41  = 1
 CrewMember2: x25  = 1
 CrewMember3: x9 + x26 + x27 + x28 + x29 + x30 + x31 + x32 + x33 + x34 + x35
              + x36 + x37 + x38  = 1
 Route0:      x1 + x11 + x13 + x15 + x16 + x17 + x19 + x20 + x21 + x22 + x23
              + x24 + x25 + x27 + x29 + x31 + x41  = 2
 Route1:      x2 + x10 + x12 + x13 + x16 + x18 + x19 + x21 + x22 + x24 + x26
              + x27 + x32 + x35 + x38 + x39 + x40 + x41  = 2
 Route2:      x3 + x14 + x15 + x17 + x20 + x23 + x25 + x28 + x29 + x30 + x31
              + x33 + x34 + x36 + x37  = 2
 Route3:      x4 + x16 + x17 + x18 + x20 + x21 + x23 + x24 + x25 + x30 + x31
              + x32 + x34 + x37 + x39 + x40 + x41  = 2
 Route4:      x5 + x19 + x20 + x21 + x22 + x23 + x24 + x25 + x33 + x34 + x35
              + x36 + x39  = 2
 Route5:      x6 + x22 + x23 + x24 + x25 + x36 + x37 + x38 + x40 + x41  = 2
 MinMax0:     x25 <= 1
 MinMax1:     x10 <= 1
 MinMax2:     x25 <= 1
 MinMax3:     x25 <= 1
 MinMax4:     x25 <= 1
 MinMax5:     x25 <= 1
```

*Figure 3.16 RME after branching in Child Problem 2.*

For the left side problem seen in Figure 3.17, the changes are minimal. Only the branching roster is removed from the problem. This is little progress compared to the right side, thus it is preferred to continue the process at the right side.

```
Minimize
 obj1: 1000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 1000000 x5
       + 1000000 x6 + 35000 x7 + 52000 x8 + 75000 x9 + 25000 x10 + 27000 x11
       + 10000 x12 + 2000 x13 + 22000 x14 + 14000 x15 + 13000 x16 + 1000 x17
       + 1000 x18 + 19000 x19 + 31000 x21 + 24000 x22 + 36000 x23 + 3000 x24
       + 44000 x25 + 27000 x26 + 19000 x27 + 39000 x28 + 31000 x29 + 10000 x31
       + 6000 x32 + 1000 x33 + 2000 x34 + 14000 x35 + 7000 x37 + 19000 x38
       + 67000 x39 + 50000 x40 + 42000 x41 + 62000 x42 + 54000 x43 + 21000 x44
       + 33000 x45 + 29000 x46 + 11000 x47 + 15000 x48 + 3000 x49 + 6000 x50
       + 2000 x53 + 12000 x54 + 4000 x55 + 3000 x56 + 11000 x57 + 15000 x58
       + 21000 x60 + 12000 x61 + 5000 x62 + 8000 x63 + 17000 x65 + 9000 x66
       + 5000 x67 + 6000 x68 + 1000 x69 + 1000 x70
Subject To
 CrewMember0: x7 + x11 + x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x20
              + x21 + x22 + x23 + x24 + x65 + x66 + x67  = 1
 CrewMember1: x8 + x25 + x26 + x27 + x28 + x29 + x30 + x31 + x32 + x33 + x34
              + x35 + x36 + x37 + x38 + x68 + x69 + x70  = 1
 CrewMember2: x9 + x39 + x40 + x41 + x42 + x43 + x44 + x45 + x46 + x47 + x48
              + x49 + x50 + x51  = 1
 CrewMember3: x10 + x52 + x53 + x54 + x55 + x56 + x57 + x58 + x59 + x60 + x61
              + x62 + x63 + x64  = 1
 Route0:      x1 + x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23
              + x24 + x25 + x27 + x29 + x30 + x31 + x33 + x34 + x35 + x36 + x37
              + x38 + x39 + x41 + x43 + x44 + x45 + x48 + x49 + x51 + x53 + x55
              + x57 + x66 + x70  = 2
 Route1:      x2 + x12 + x13 + x16 + x21 + x23 + x26 + x27 + x30 + x32 + x33
              + x35 + x36 + x38 + x40 + x41 + x44 + x46 + x47 + x49 + x50 + x51
              + x52 + x53 + x58 + x61 + x64 + x67 + x68 + x69 + x70  = 2
 Route2:      x3 + x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 + x28 + x29
              + x31 + x34 + x37 + x42 + x43 + x45 + x48 + x54 + x55 + x56 + x57
              + x59 + x60 + x62 + x63 + x65 + x66  = 2
 Route3:      x4 + x16 + x17 + x18 + x19 + x21 + x22 + x23 + x30 + x31 + x32
              + x34 + x35 + x37 + x38 + x44 + x45 + x46 + x47 + x48 + x49 + x50
              + x51 + x56 + x57 + x58 + x60 + x63 + x68 + x69 + x70  = 2
 Route4:      x5 + x19 + x20 + x21 + x22 + x23 + x24 + x33 + x34 + x35 + x36
              + x37 + x38 + x47 + x48 + x49 + x50 + x51 + x59 + x60 + x61 + x62
              + x68  = 2
 Route5:      x6 + x22 + x23 + x24 + x36 + x37 + x38 + x50 + x51 + x62 + x63
              + x64 + x65 + x66 + x67 + x69 + x70  = 2
 MinMax0:     x11 + x13 + x15 + x16 + x18 + x19 + x20 + x21 + x22 + x23 + x24
              + x39 + x41 + x43 + x44 + x45 + x48 + x49 + x51 + x66 <= 1
 MinMax1:     x12 + x13 + x16 + x21 + x23 + x40 + x41 + x44 + x46 + x47 + x49
              + x50 + x51 + x67 <= 1
 MinMax2:     x14 + x15 + x17 + x18 + x19 + x20 + x22 + x24 + x42 + x43 + x45
              + x48 + x65 + x66 <= 1
 MinMax3:     x16 + x17 + x18 + x19 + x21 + x22 + x23 + x44 + x45 + x46 + x47
              + x48 + x49 + x50 + x51 <= 1
 MinMax4:     x19 + x20 + x21 + x22 + x23 + x24 + x47 + x48 + x49 + x50 + x51
              <= 1
 MinMax5:     x22 + x23 + x24 + x50 + x51 + x65 + x66 + x67 <= 1
```

*Figure 3.17 RME after branching in Child Problem 1.*

After branching the column generation is called again to add promising rosters to the altered restricted masted problem. The column generation adds promising variables x42 and x41 as seen in Figure 3.18. No other roster is available for crew member 0 as for pairings 0, 2, 3, 4, and 5 crew member 2 is assigned and the Min-Max limit does not allow more crew members with the characteristic.

```
Minimize
 obj1: 1000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 1000000 x5
       + 1000000 x6 + 35000 x7 + 52000 x8 + 25000 x9 + 10000 x10 + 44000 x11
       + 27000 x12 + 19000 x13 + 39000 x14 + 31000 x15 + 10000 x17 + 6000 x18
       + 1000 x19 + 2000 x20 + 14000 x21 + 7000 x23 + 19000 x24 + 10000 x25
       + 2000 x27 + 12000 x28 + 4000 x29 + 3000 x30 + 11000 x31 + 15000 x32
       + 21000 x34 + 12000 x35 + 5000 x36 + 8000 x37 + 6000 x39 + 1000 x40
       + 1000 x41
Subject To
 CrewMember0: x7 + x10  = 1
 CrewMember1: x8 + x11 + x12 + x13 + x14 + x15 + x16 + x17 + x18 + x19 + x20
              + x21 + x22 + x23 + x24 + x39 + x40 + x41  = 1
 CrewMember2: x25  = 1
 CrewMember3: x9 + x26 + x27 + x28 + x29 + x30 + x31 + x32 + x33 + x34 + x35
              + x36 + x37 + x38 + x42 + x43  = 1
 Route0:      x1 + x11 + x13 + x15 + x16 + x17 + x19 + x20 + x21 + x22 + x23
              + x24 + x25 + x27 + x29 + x31 + x41 + x42  = 2
 Route1:      x2 + x10 + x12 + x13 + x16 + x18 + x19 + x21 + x22 + x24 + x26
              + x27 + x32 + x35 + x38 + x39 + x40 + x41  = 2
 Route2:      x3 + x14 + x15 + x17 + x20 + x23 + x25 + x28 + x29 + x30 + x31
              + x33 + x34 + x36 + x37 + x42  = 2
 Route3:      x4 + x16 + x17 + x18 + x20 + x21 + x23 + x24 + x25 + x30 + x31
              + x32 + x34 + x37 + x39 + x40 + x41 + x43  = 2
 Route4:      x5 + x19 + x20 + x21 + x22 + x23 + x24 + x25 + x33 + x34 + x35
              + x36 + x39  = 2
 Route5:      x6 + x22 + x23 + x24 + x25 + x36 + x37 + x38 + x40 + x41 + x42
              + x43  = 2
 MinMax0:     x25 <= 1
 MinMax1:     x10 <= 1
 MinMax2:     x25 <= 1
 MinMax3:     x25 <= 1
 MinMax4:     x25 <= 1
 MinMax5:     x25 <= 1
```

*Figure 3.18 RME after second complete iteration of Column Generation.*

As the column generation has finished, an integral solution has been found which fits the requirements. Thus, the methodology has been completed.

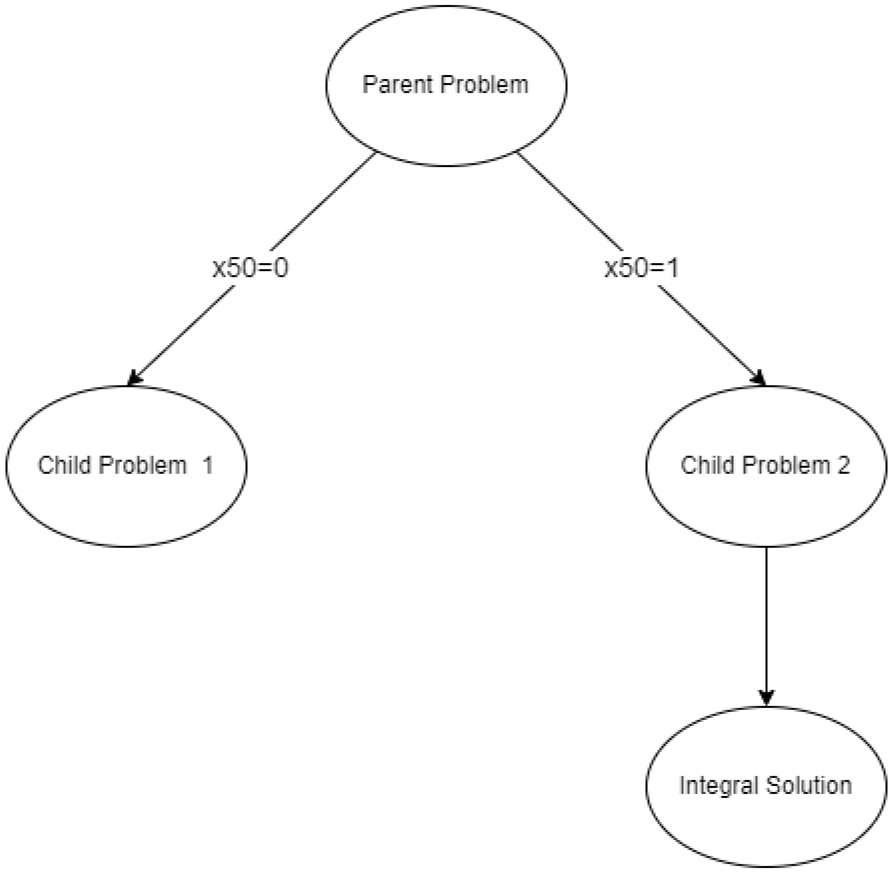


*Figure 3.19 Finding an integral solution.*

The solution objective function is 21.000 with crew member 0 been assigned to pairing 1, crew member 1 to pairings 0, 1, 3, and 5, crew member 2 to pairings 0, 2, 3, 4, and 5, and crew member 3 been assigned to pairings 2 and 4. The cost in the objective function originates from ten hours deviation from the desired window of crew member 0, from one hour from crew member 1, and other ten hours from crew member 2.

# 4 Computational Study

In the previous chapter two master problems were illustrated with two variations, with or without the Min-Max constraints. To solve these formulations, a branch and price methodology was presented and tested in an example. To evaluate the performance of the methodology, a program has been developed in C programming language using IBM ILOC CPLEX library to solve the set cover and set partition with both of their variations. Six sets of different set ups have been created for this task and the results will be compared between themselves. The tests where run on a personal computer with 16GB of RAM and an Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz CPU.

## 4.1 Setting the Scenarios

To evaluate the program, six scenarios have been designed. The base is three scenarios where each of the base scenarios has a two variation of a high and low utilization of the crew member. The low utilization scenarios have an average of four pairings to each crew member; on the other hand, the high utilization has an average of five pairings per crew member. Each scenario has five instances with a growing number of crew members, 50, 100, 150, 200, and 250 accordingly. The number of crew members of the scenarios are in numbers of a small or medium airline depending also on whether the crew rostering problem is run per crew member category/position or not.

The pairing data and crew data have been generated randomly; the generated values are according to the regulation where applicable. For the pairings, the flight time and date in a duration of a month is generated, further than that, the duration of each pairing is generated according to regulations. For the crew members, a time window of preferred total flight time is created respecting the rules that govern the total work time. In addition, a Boolean is created based on a probability $p$ for each crew member regarding the characteristic which will play role in the min-max constraints. The probability $p$ is stated in each scenario as a parameter to evaluate its effect on the calculations.

The MIP Gap Tolerance is set equal to 5.000. This is the minimum absolute difference between the optimal objective value of an integer solution and the optimal objective value of the linear problem for which the program continues to find a better integer solution. If the difference is less than the parameter, the program terminates. The corresponding percentage value is equal to 10 %. This is the minimum percentage difference between the optimal objective value of an integer solution and the optimal objective value of the linear problem for which the program continues to find a better integer solution. If the difference is less than the parameter, the program terminates. Also, the maximum number of backtrack is set to 100 and the maximum number of tree nodes created is set equal to 20*N.

For the legality of a roster, as the period of scheduling time is set to one month, the lower limit for crew members days off is set to 10. In addition, maximum flight hours cannot be over 100.

To compere the performance, we used the same generated data in an instance and run the program four times to cover all the formulations, set cover, set partition, set cover with Min-Max constraints, set partition with Min-Max constraints.

## 4.2 Results

### 4.2.1 Scenario 1 – Balanced and low utilization

For this scenario and its instances, the parameters used are seen in Table 4.1. When the number of crew members are increased, so do the pairings increase to fit the load of average four pairing per crew member in the low utilization scenarios. The complement of each pairing remains the same for the entire scenario and is equal to two. The probability $p$ determines whether a crew member has the characteristic limited in the Min-Max constraints. The limit of the Min-Max constraints in these instances is the parameter $d$ shown in the table.

*Table 4.1 Scenario 1 Parameters.*

| Scenarios | **N:** Number of Crew Members | **R:** Number of Pairings | **b:** Pairings Complements | **d:** Min Max | **P:** Possibility to have the Characteristic |
|---|---|---|---|---|---|
| S1.1 | 50 | 100 | 2 | 1 | 50% |
| S1.2 | 100 | 200 | 2 | 1 | 50% |
| S1.3 | 150 | 300 | 2 | 1 | 50% |
| S1.4 | 200 | 400 | 2 | 1 | 50% |
| S1.5 | 250 | 500 | 2 | 1 | 50% |

*Table 4.2 Scenario 1 Computational results.*

| Scenarios | Without Min Max constraints | | | | | | With Min Max Constraints | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. |
| S1.1 | 0:00:02 | 0 | 0 | 0:01:07 | 3.000 | 0 | 0:00:10 | 0 | 0 | 0:03:52 | 6.000 | 0 |
| S1.2 | 0:01:37 | 0 | 0 | 0:09:06 | 91.000 | 0 | 0:02:22 | 0 | 0 | 0:13:15 | 97.000 | 0 |
| S1.3 | 0:07:24 | 0 | 0 | 0:34:13 | 376.000 | 0 | 0:15:25 | 0 | 0 | 1:27:42 | 875.000 | 0 |
| S1.4 | 0:11:44 | 0 | 0 | 1:18:21 | 277.000 | 0 | 0:45:32 | 0 | 0 | 2:48:32 | 751.000 | 0 |
| S1.5 | 1:12:31 | 0 | 0 | 3:03:52 | 880.000 | 0 | 1:30:11 | 0 | 0 | 4:15:34 | 880.000 | 0 |

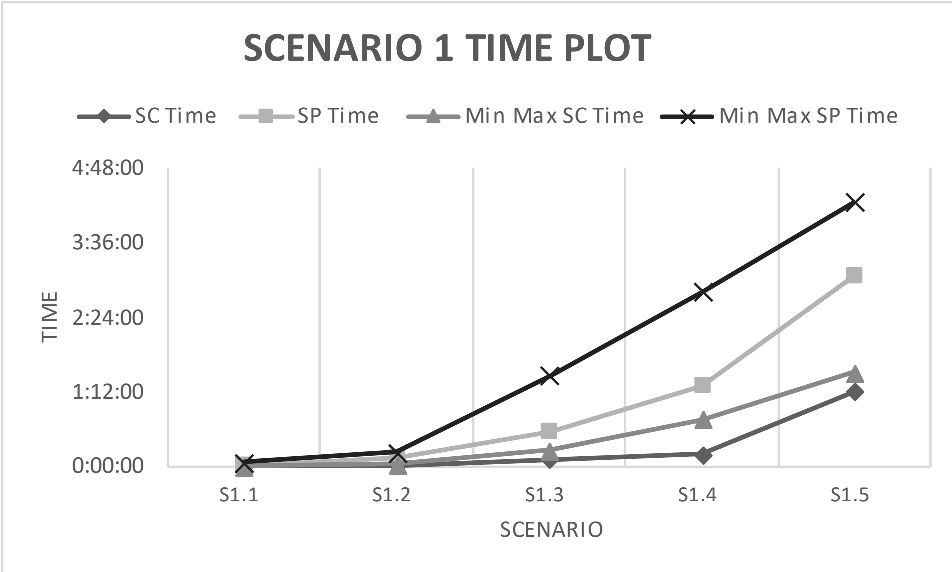


*Figure 4.1 Time Comparison plot for scenario 1.*

In this scenario, the crew members are mostly lightly utilized, meaning in the end of the month most of them have total flight time less that their preferred window and have more hours available to fill any needed pairing. The set partition aims to give perfect fit roster to as many crew members to reduce the total quality cost. The Min-Max constraints pose a greater challenge to the balancing of the rosters, resulting in lower roster quality for the crew members and higher total cost. The instances S1.1, S1.2, and S1.5 seem relative reasonable with the addition of the constraints but instances S1.3 and S1.4 were affected more. Due to the light schedule, all pairings were covered in all instances. The time needed to complete the min-max problems was higher than the problems without min-max constraints. Though higher, the time is still comparable.

### 4.2.2 Scenario 2 - Balanced and high utilization

*Table 4.3 Scenario 2 Parameters.*

| Scenarios | **N:** Number of Crew Members | **R:** Number of Pairings | **b:** Pairings Complements | **d:** Min Max | **P:** Possibility to have the Characteristic |
|---|---|---|---|---|---|
| S2.1 | 50 | 125 | 2 | 1 | 50% |
| S2.2 | 100 | 250 | 2 | 1 | 50% |
| S2.3 | 150 | 375 | 2 | 1 | 50% |
| S2.4 | 200 | 500 | 2 | 1 | 50% |
| S2.5 | 250 | 625 | 2 | 1 | 50% |

*Table 4.4 Scenario 2 Computational results.*

| Scenarios | Without Min Max constraints | | | | | | With Min Max Constraints | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. |
| S2.1 | 0:00:17 | 0 | 0 | 0:00:53 | 489.000 | 0 | 0:00:44 | 0 | 0 | 0:02:06 | 489.000 | 0 |
| S2.2 | 0:03:53 | 0 | 0 | 0:09:35 | 1.075.000 | 0 | 0:11:35 | 0 | 0 | 0:17:43 | 1.076.000 | 0 |
| S2.3 | 0:14:20 | 0 | 0 | 0:40:55 | 1.333.000 | 0 | 0:52:17 | 1.000.000 | 1 | 3:50:09 | 1.333.000 | 0 |
| S2.4 | 0:59:55 | 0 | 0 | 2:27:34 | 1.640.000 | 0 | 1:09:55 | 0 | 0 | 3:58:30 | 1.634.000 | 0 |
| S2.5 | 2:28:24 | 0 | 0 | 7:05:44 | 2.565.000 | 0 | 3:44:23 | 0 | 0 | 20:05:54 | 2.576.000 | 0 |

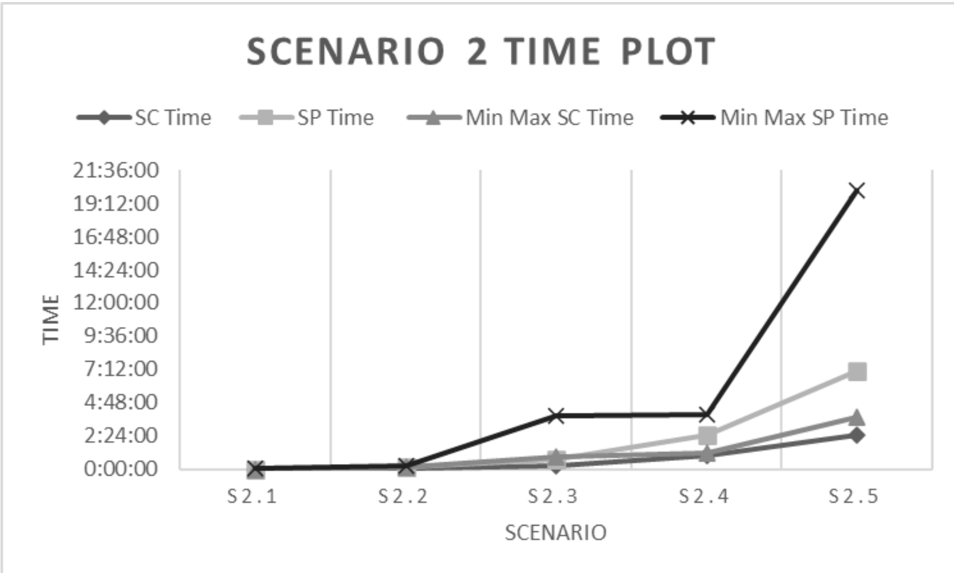


*Figure 4.2 Time Comparison plot for scenario 2.*

This scenario is similar to scenario 1 with the difference that the crew members are highly utilized. In both set partition solutions most of the pilots were over their time window meaning that in these instances the covering of the pairings is at stake. Both set partition problems were able to cover all pairings while the set cover solutions proved that such a solution was available. The solutions of set partition were very close, due to not being able to achieve the desired window for both cases resulting in similar objective functions. Something not expected occurred in instance S2.4 where the achieved solution of the set partition problem with Min-Max was better that the problem without, this may be due to an early branching which lost some potential. Regarding time, the calculation time for the set partition problem with the Min-Max is significant more than the problem without. For the set cover formulations, the results are comparable except for instance S2.3 where a bad branching resulting on an uncovered pairing. In the set partition with min-max, all pairings were covered meaning that there is an available schedule to cover all pairings legally. This points out that the branching strategy may need revaluation to protect against such cases.

### 4.2.3 Scenario 3 - Unbalanced and low utilization

*Table 4.5 Scenario 3 Parameters.*

| Scenarios | **N**: Number of Crew Members | **R**: Number of Pairings | **b**: Pairings Complements | **d:** Min Max | **P**: Possibility to have the Characteristic |
|---|---|---|---|---|---|
| S3.1 | 50 | 100 | 2 | 1 | 60% |
| S3.2 | 100 | 200 | 2 | 1 | 60% |
| S3.3 | 150 | 300 | 2 | 1 | 60% |
| S3.4 | 200 | 400 | 2 | 1 | 60% |
| S3.5 | 250 | 500 | 2 | 1 | 60% |

*Table 4.6 Scenario 3 Computational results.*

| Scenarios | Without Min Max constraints | | | | | | With Min Max Constraints | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. |
| S3.1 | 0:00:02 | 0 | 0 | 0:01:07 | 3.000 | 0 | 0:00:13 | 0 | 0 | 0:01:20 | 526.000 | 0 |
| S3.2 | 0:01:37 | 0 | 0 | 0:09:06 | 91.000 | 0 | 0:02:39 | 0 | 0 | 0:16:47 | 106.000 | 0 |
| S3.3 | 0:07:24 | 0 | 0 | 0:34:13 | 376.000 | 0 | 0:32:44 | 0 | 0 | 0:47:12 | 2.888.000 | 0 |
| S3.4 | 0:11:44 | 0 | 0 | 1:18:21 | 277.000 | 0 | 0:44:22 | 0 | 0 | 2:34:24 | 2.393.000 | 0 |
| S3.5 | 1:12:31 | 0 | 0 | 3:03:52 | 880.000 | 0 | 2:01:27 | 0 | 0 | 5:47:31 | 2.614.000 | 0 |

In this scenario, the crew members with the characteristic are chosen to be slightly more, somewhere around 60%, as that is the possibility to be a crew member with the characteristic. Again, the crew schedule is made to be a light schedule which gives freedom to the program to achieve a better quality of rosters.


*Figure 4.3 Time Comparison plot for scenario 3.*

Reviewing the results, the unbalance brought high cost to the set partition with min-max formulation, while on the other hand all the pairings were able to be covered as stated in the corresponding set cover formulation. The time needed to achieve a solution for both cases is increased but is reasonable.

### 4.2.4 Scenario 4 - Unbalanced and high utilization

*Table 4.7 Scenario 4 Parameters.*

| Scenarios | **N**: Number of Crew Members | **R**: Number of Pairings | **b**: Pairings Complements | **d**: Min Max | **P**: Possibility to have the Characteristic |
|---|---|---|---|---|---|
| S4.1 | 50 | 125 | 2 | 1 | 60% |
| S4.2 | 100 | 250 | 2 | 1 | 60% |
| S4.3 | 150 | 375 | 2 | 1 | 60% |
| S4.4 | 200 | 500 | 2 | 1 | 60% |
| S4.5 | 250 | 625 | 2 | 1 | 60% |

*Table 4.8 Scenario 4 Computational results.*

| Scenarios | Without Min Max constraints | | | | | | With Min Max Constraints | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. |
| S4.1 | 0:00:17 | 0 | 0 | 0:00:53 | 489.000 | 0 | 0:01:29 | 16.000.000 | 16 | 0:02:18 | 11.460.000 | 11 |
| S4.2 | 0:03:53 | 0 | 0 | 0:09:35 | 1.075.000 | 0 | 0:02:28 | 0 | 0 | 0:23:27 | 1.081.000 | 0 |
| S4.3 | 0:14:20 | 0 | 0 | 0:40:55 | 1.333.000 | 0 | 0:49:05 | 17.000.000 | 17 | 1:20:14 | 19.611.000 | 18 |
| S4.4 | 0:59:55 | 0 | 0 | 2:27:34 | 1.640.000 | 0 | 2:09:43 | 42.000.000 | 42 | 3:43:58 | 62.568.000 | 61 |
| S4.5 | 2:28:24 | 0 | 0 | 7:05:44 | 2.565.000 | 0 | 8:19:56 | 85.000.000 | 85 | 12:04:37 | 5.919.000 | 3 |

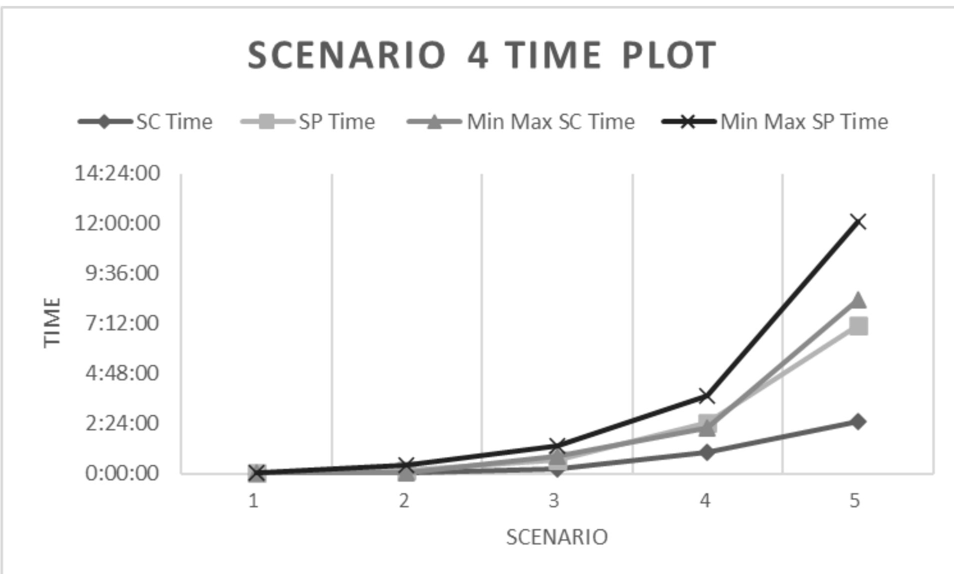


*Figure 4.4 Time Comparison plot for scenario 4.*

Reviewing this scenario, it is seen that the covering of all pairings probably was not possible with the Min-Max as not all crew members could be utilized for all pairing. In Figure 4.4, it is seen that the time needed to reach a solution is much higher for the formulations with Min-Max. This is also the only scenario for which the set cover formulation with Min-Max needs more time to conclude to a solution than the set partition formulation without Min-Max. It is also seen again that the branching strategy has weakness in the formulation with Min-Max as we see solutions of set cover with more uncovered pairings that the set partition one.

### 4.2.5 Scenario 5 – Flexible and low utilization

*Table 4.9 Scenario 5 Parameters.*

| Scenarios | **N**: Number of Crew Members | **R**: Number of Pairings | **b**: Pairings Complements | **d:** Min Max | **P**: Possibility to have the Characteristic |
|---|---|---|---|---|---|
| S5.1 | 50 | 66 | 3 | 2 | 50% |
| S5.2 | 100 | 133 | 3 | 2 | 50% |
| S5.3 | 150 | 200 | 3 | 2 | 50% |
| S5.4 | 200 | 266 | 3 | 2 | 50% |
| S5.5 | 250 | 333 | 3 | 2 | 50% |

In this scenario the program is put to test an easier fit regarding the Min-Max constraints. The crew members with the characteristic are at 50% but the limit is two crew members to a pairing with a complement of three. Again, this scenario is for the low utilization.

*Table 4.10 Scenario 5 Computational results.*

| Scenarios | Without Min Max constraints | | | | | | With Min Max Constraints | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. |
| S5.1 | 0:00:03 | 0 | 0 | 0:00:18 | 59.000 | 0 | 0:00:02 | 0 | 0 | 0:00:31 | 59.000 | 0 |
| S5.2 | 0:00:07 | 0 | 0 | 0:03:30 | 179.000 | 0 | 0:00:58 | 0 | 0 | 0:03:55 | 179.000 | 0 |
| S5.3 | 0:04:43 | 0 | 0 | 0:13:25 | 78.000 | 0 | 0:04:53 | 0 | 0 | 0:14:58 | 78.000 | 0 |
| S5.4 | 0:03:28 | 0 | 0 | 0:37:07 | 425.000 | 0 | 0:19:07 | 0 | 0 | 0:45:24 | 425.000 | 0 |
| S5.5 | 0:24:04 | 0 | 0 | 1:30:19 | 628.000 | 0 | 0:44:32 | 0 | 0 | 1:38:46 | 628.000 | 0 |


*Figure 4.5 Time Comparison plot for scenario 5.*

As shown in the comparison plot above, the variations needed only a slight difference in the computational time. The solution achieved the same objective function as the limit was not a tight one. The computational time needed for this scenario is much less than the previous ones, this has to do also with the fact that we are covering the same average number of pairings per crew member but with less pairings in total due to higher number of complements. This results in a much smaller network used in the column generation, which impacts positively the computational time.

### 4.2.6 Scenario 6 – Flexible and high utilization

*Table 4.11 Scenario 6 Parameters.*

| Scenarios | **N**: Number of Crew Members | **R**: Number of Pairings | **b**: Pairings Complements | **d:** Min Max | **P**: Possibility to have the Characteristic |
|---|---|---|---|---|---|
| S6.1 | 50 | 83 | 3 | 2 | 50% |
| S6.2 | 100 | 166 | 3 | 2 | 50% |
| S6.3 | 150 | 250 | 3 | 2 | 50% |
| S6.4 | 200 | 333 | 3 | 2 | 50% |
| S6.5 | 250 | 416 | 3 | 2 | 50% |

*Table 4.12 Scenario 6 Computational results.*

| Scenarios | Without Min Max constraints | | | | | | With Min Max Constraints | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. | SC Time | SC Cost | SC Uncov. | SP Time | SP Cost | SP Uncov. |
| S6.1 | 0:00:08 | 0 | 0 | 0:00:20 | 627.000 | 0 | 0:00:10 | 0 | 0 | 0:00:22 | 627.000 | 0 |
| S6.2 | 0:01:52 | 0 | 0 | 0:04:43 | 965.000 | 0 | 0:02:24 | 0 | 0 | 0:05:04 | 965.000 | 0 |
| S6.3 | 0:10:14 | 0 | 0 | 0:17:58 | 1.445.000 | 0 | 0:13:07 | 0 | 0 | 0:21:56 | 1.445.000 | 0 |
| S6.4 | 0:24:01 | 0 | 0 | 0:54:10 | 1.623.000 | 0 | 0:40:16 | 0 | 0 | 1:09:43 | 1.623.000 | 0 |
| S6.5 | 1:06:48 | 0 | 0 | 2:00:30 | 2.049.000 | 0 | 1:23:58 | 0 | 0 | 2:13:28 | 2.049.000 | 0 |


*Figure 4.6 Time Comparison plot for scenario 6.*

In Scenario 6, the solution time is very close for the variations and the objective functions achieved is the same. Thus, the program may bring solutions respecting the Min-Max constraints with no disadvantage.

# 5 Summary

In the present thesis, a branch and price methodology was developed to address the crew rostering problem. The crew rostering problem is a known part of the crew scheduling for the airline industry, which aims to assign each crew member to a set of pairings, with pairings being a set of flight legs which were created by solving the crew pairing problem. The problem is of great interest to the operation researchers due to its complexity and sheer size. The complexity originating from a set of tight regulations which dictate the legality of the rosters and the size due to the large fleets of the airlines which partake in many flights with a large number of crew members required.

In the formulation of the problems, we have taken into consideration a set of rules for the assignment of crew members to pairings based on a maximum number of crew members with a specific characteristic for each pairing. This is handled through the addition of a set of specific constraints, called Min-Max constraints. A small trick is presented for the case in which a minimum number of crew members with a special characteristic is needed.

To solve the formulated problems a branch and price methodology was implemented. During the column generation phase, the legality of the rosters was checked based on a regulation scheme which mimics reality. On the branch and bound phase, we set rules to efficiently move around the subproblems moving fast to a near optimal solution.

To evaluate the proposed formulation of the crew roster problem, a program was developed using C programming language and IBM ILOG CPLEX libraries. Then multiple scenarios were tested using the set cover and set partition formulations, with and without the Min-Max constraints.

By comparing the results, the conclusion was that a close to optimal solution is possible but at an increase of computational resources. The branching progress may be slightly adjusted in order to not follow branching routes which may seem the most promising at the point of branching but result in lower quality solutions.

## 5.1 Future work

As this thesis tries to further the research of years of operations research in the airline industry, so this work may be further researched.

We have stated that the branching strategy works fast and finds a near optimal solution, especially in the solutions of the formulated problems without the Min-Max constraints. But in the case of the problems with the Min-Max constraints branching may follow a route of branching which results in worse solutions, possible with not needed uncovered pairings and underutilized crew members. With a better approach, faster and improved solutions may be found.

Further than the branching strategy, also other parameters may be adjusted to alternate the performance of the program. Depending on the parameters, the program may bring a solution with less computational resources or improve the solutions at the expense of computational resources.

Depending on the problem and the targets set by the schedule planner the parameters should be adjusted accordingly.

Something not researched in the scope of this thesis is the influence on the solution if instead of one set of Min-Max constraints they were more. As the regulations are plenty and different per country, while the crew members also have many different sets of characteristics, the Min-Max constraints may be used for many objectives. Thus, more than one set of Min-Max constraints could be implemented in the formulation of the problem.

In addition, there are mathematical models to transform from the set cover to the set partition solution as developed in [26]. This two-step solution, set cover and then set cover to set partition, may not produce a better solution than set partition but it may provide a solution in a better time. Another benefit may lay if the solution given by the two-step solution is used as the restricted master problem in the set partition problem using the branch and price methodology developed.

Finally, the greatest way to a evaluate a methodology is by testing its performance in real cases. Feeding the problem with real data will lead to better adjustment of the parameters set in this methodology and present any weakness in the development which needs to be addressed.

# 6 Bibliography

[1] A. T. A. o. America, ATA Airline Handbook, 2001.

[2] M. F. Scharpenseel, “Consequences of eu airline deregulation in the context of the global aviation market,” *Northwestern Journal of International Law & Business,* vol. 22, no. 1, 2001.

[3] G. g. a. I. High-level, “Aviation Benefits Report 2019,” Industry High Level Group, 2019.

[4] K. Kiraci, “THE IMPACT OF THE GLOBAL FINANCIAL CRISIS ON AIRLINES,” in *FINANCIAL RISK & FINANCIAL PERFORMANCE:,* London, IJOPEC Publication Limited, 2019, pp. 257-284.

[5] P. B. A. R. O. Cynthia Barnhart, “Applications of Operations Research in the Air Transport Industry,” *Transportation Science,* vol. 37, pp. 368-391, 2003.

[6] J. F. L. a. R. M. D. Barry C. Smith, “Yield Management at American Airlines,” *Interfaces,* vol. 22, no. 1, pp. 8-31, 1992.

[7] C. Barnhart, A. M. Cohn, E. L. Johnson, D. Klabjan, G. L. Nemhauser and P. H. Vance, “Airline Crew Scheduling,” in *Handbook of Transportation Science,* Boston, Springer, 2004, pp. 517-560.

[8] N. KOHL and S. E. KARISCH, “Airline Crew Rostering: Problem Types, Modeling, and Optimization,” *Annals of Operations Research,* vol. 127, pp. 223-257, 2004.

[9] G. W. Graves, R. D. McBride, I. Gershkoff, D. Anderson and D. Mahidhara, “Flight Crew Scheduling,” *Management Science,* vol. 39, no. 6, pp. 736-745, 1993.

[10] F. Quesnel, G. Desaulniers and F. Soumis, “A branch-and-price heuristic for the crew pairing problem with language constraints,” *European Journal of Operational Research,* vol. 283, no. 3, 2019.

[11] G. Desaulniers, D. J., Y. Dumas, S. Marc, B. Rioux, M. M. Solomon and F. Soumis, “Crew Pairing at Air France.,” *European Journal of Operational Research,* vol. 97, p. 245–259, 1997.

[12] F. Quesnel, G. Desaulniers and F. Soumis, “A New Heuristic Branching Scheme for the Crew Pairing Problem with Base Constraints,” *Computers & Operations Research,* vol. 80, p. 159–172, 2017.

[13] P. H. Vance, C. Barnhart, E. L. Johnson and G. L. Nemhauser, “Airline Crew Scheduling: A New Formulation and Decomposition Algorithm,” *Operations Research,* vol. 45, p. 188–200, 1997.

[14] B. Zeren and I. Özkol, “A Novel Column Generation Strategy for Large Scale Airline Crew Pairing Problems.,” *Expert Systems with Applications,* vol. 55, pp. 133-144, 2016.

[15] B. GOPALAKRISHNAN and E. L. JOHNSON, “Airline Crew Scheduling: State-of-the-Art,” *Annals of Operations Research,* vol. 140, p. 305–337, 2005.

[16] I. T. Christou, A. Zakarian, J.-M. Liu and H. Carter, “A Two-Phase Genetic Algorithm for Large Scale Bidline-Generation Problems at Delta Air Lines.,” *Interfaces,,* vol. 29, pp. 51-65, 1999.

[17] M. Gamache, F. Soumis, D. Villeneuve, J. Desrosiers and É. Gélinas, “The Preferential Bidding System at Air Canada,” *Transportation Science,* vol. 32, p. 246–255, 1998.

[18] Boubaker, D. G. K. and I. Elhallaoui, “Bidline Scheduling with Equity by Heuristic Dynamic Constraint Aggregation,” *Transportation Research Part B,,* vol. 44, p. 50–61, 2010.

[19] J. de Armas, L. Cadarso, A. A. Juan and J. Faulin, “A Multi-Start Randomized Heuristic for Real-Life Crew Rostering Problems in Airlines with Work-Balancing Goals,” *Annals of Operations Research,* vol. 258, p. 825–848, 2017.

[20] M. Gamache, F. Soumis, G. Marquis and J. Desrosiers, “A Column Generation Approach for Largescale Aircrew Rostering Problems,” *Operations Research,* vol. 47, p. 247–263, 1999.

[21] C. P. Medard and N. Sawhney, “Airline Crew Scheduling from Planning to Operations,” *European Journal of Operational Research,* vol. 183, p. 1013–1027, 2007.

[22] A. Kasirzadeh, M. Saddoune and F. Soumis, “Airline Crew Scheduling: Models, Algorithms, and Data Sets.,” *EURO Journal on Transportation and Logistics*, vol. 6, p. 111–137, 2017.

[23] H. Kharraziha, M. Ozana and S. Spjuth, “Large Scale Crew Rostering.,” Goteborg, 2003.

[24] Y.-D. Shen and C. Shijun, “A column generation algorithm for crew scheduling with multiple additional constraint,” *Pacific Journal of Optimization* , vol. 10, 2014.

[25] B. Maenhout and M. Vanhoucke, “A Hybrid Scatter Search Heuristic for Personalized Crew Rostering in the Airline Industry,” *European Journal of Operational Research*, vol. 206, p. 155–167, 2010.

[26] O. C. Moschopoulos, *Set-partition versus set-cover formulations for column generation based solution methodologies for optimal crew scheduling in the airline industry,* Volos: University of Thessaly, 2018.

[27] M. Akella, A. Sarkar and S. Gupta, *Branch and Price Column Generation for Solving Huge Integer Programs*, Univerity of Buffalo, 2011.