Runway Operations Scheduling using Airline Preferences

Soomer, Maarten Jan, 1978
Runway Operations Scheduling using Airline Preferences
ISBN 978-90-8659-262-3

The research in this thesis has been supported by the National Aerospace Laboratory NLR.

© M.J. Soomer, Amsterdam 2008.
Cover design by Iris Koene (NLR).

All rights reserved. No part of this publication may be reproduced in any form or by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval systems) without permission in writing from the author.

Printed by PrintPartners Ipskamp, The Netherlands.

Thomas Stieltjes Institute
for Mathematics

VRIJE UNIVERSITEIT

Runway Operations Scheduling using Airline Preferences

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. L.M. Bouter,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op vrijdag 9 januari 2009 om 10.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Maarten Jan Soomer

geboren te Amstelveen

promotor: prof.dr. G.M. Koole

Preface

It's done! This thesis is the result of the research I have been performing the past years. During this period I was working both at the VU University and the Dutch National Aerospace Laboratory NLR. Further I visited interesting conferences at nice locations, such as San Francisco, Denver and Belgrade. The work also included supervising students for projects, master theses and business cases, which I very much enjoyed.

Here I want to thank the people that provided support, cooperation and motivation during the research.

First of all, I want to thank my supervisor Ger Koole. He is a driven professor, who trusts on the independence of the researcher and is always open to opportunities. I am very thankful for the opportunity he gave me to spend 6 months in the USA working on a very interesting project at Park 'n Fly. This was a great experience and a pleasant break from my PhD research.

I want to thank the reading committee for their useful corrections, comments and suggestions. These helped to improve the thesis considerably. The reading committee consists of Thomas Vossen, Jeroen Mulder, Michel Van Eenige, Geert-Jan Franx and Rob van der Mei.

Geert-Jan Franx also provided useful contributions during the research, especially on the subject of airline cost representation and related fairness issues.

I am thankful to Bart van Asten and Jeroen Mulder for providing input from the airline point of view.

Thanks go also out to Rene Verbeek, Michel van Eenige and Ronny Groothuizen from the NLR for their input to the research.

Working at the university was a pleasant experience. The OBP research group forms a group of nice and enthusiastic people. I also enjoyed the lunches we had together with a group of very nice topologists. The Thursday afternoon drinks, barbecues and other activities were a good way to get to know colleagues more personally.

Fellow PhD students, Menno and Auke became good friends and will be my *paranimfen* at the defense. Together we saved the world several times while enjoying some beers and talking German at ski vacations.

It was always good to blow off some steam at the end of the week and start the weekend with Karin, Jeroen, Bram, Petra and all the others at Friday after work at the Gambrinus. I hope this tradition will last for a long time.

Wouter is another good friend I would like to thank for his interest and support during these years.

Finally, I want to thank my mother for her unconditional support and love.

Maarten Jan Soomer
Amsterdam, November 2008

Contents

Chapter 1

Introduction

The amount of air traffic has been increasing for decades. After an interruption caused by the September 11 attacks in 2001, this trend continues. Within Europe there has been a 23% increase in the number of flights in the period from 2002 to 2007 [38]. This growth is not being met by a corresponding increase in the physical capacity of the air traffic system, such as new airports and runways. This often leads to congestion and delays in the air traffic system.

At the same time, many airlines are struggling to survive. Nowadays, large capital investments are required to operate an airline. Operational costs are high, especially with the current fuel prices. Security measures have been increased since the September 11 attacks, resulting in additional cost. Next to that, competition is fierce. Due to low-cost airlines, (leisure) travelers have grown accustomed to low ticket prices. All this has forced airlines to work very efficiently. Advanced operations research models are used by airlines to consider efficiency in virtually every decision from defining timetables and scheduling crew and aircraft to selling seats for the right price.

However, the delays caused by air traffic congestion are largely beyond the control of airlines, because air traffic is managed by Air Traffic Control organizations and procedures. Yet, these delays have an enormous impact on the cost of an airline. If a flight has to fly a longer route or has to wait before landing, additional fuel is used. Because of delays the original crew and aircraft assignments can become impractible. In this way, delays are propagated to subsequent flights. It might even be necessary to cancel flights. Of course, all this leads to additional costs, such as crew overtime payments.

Large delays can be very frustrating to passengers. Consider for example the bad press JetBlue Airways received for forcing hundreds of passengers to sit aboard grounded airplanes for up to 10 hours because of weather-related delays in New York City in February 2007, even leading to discussion in the U.S. congress [2]. Although this is an extreme case, airline delays do have a negative effect on passenger goodwill.

Delay is especially troublesome if it results in passengers missing connecting flights. In this case, passengers have to be rebooked on other flights and sometimes have to be compensated by the airline. When large numbers of flights are delayed, it can be difficult to accommodate all transfer passengers. Sometimes overnight stays have to be arranged, because no later flight is available on the same day.

All this shows that air traffic delays have an enormous impact on airlines and their passengers. However, the impact of a delay will differ from flight to flight, depending, among others, on the number of (transfer) passengers. An airline will often prefer a delay for a flight without any transfer passengers over a delay for a flight full of time-critical transfer passengers. It is expected that by considering these preferences in air traffic control decisions the impact of delay on the airlines and their passengers can be reduced. This will lead to cost savings for airlines and fewer frustrations for passengers.

A difficulty when considering individual airlines preferences in Air Traffic Control decisions is fairness. It is the role of air traffic control to assure that air traffic proceeds in a safe, efficient and equitable manner. Consequently, scarce air traffic capacity has to be assigned to competing airlines in a fair manner.

The purpose of the research in this thesis is to explore the effects of considering airline preferences in air traffic control decisions. The fairness issues that stem from this are explicitly considered in the research.

1.1 Runway Operations Scheduling

In the research, runway operations scheduling is the air traffic decision problem that is considered. This problem involves the scheduling of landings and take-offs at runways. In this section we will motivate this choice and provide a further introduction to runway operations scheduling.

As mentioned in the previous section, air traffic volumes have been increasing for decades. Significant improvements have already been achieved in enlarging the en-route traffic throughput. As a result, airports form nowadays a bottleneck in the air traffic system. The capacity of an airport is mainly determined by the runway system capacity, i.e., the maximum number of take-offs and landings that can take place in a certain time period.

Flights have to be separated sufficiently at the runway for reasons of safety. The separation distances are dependent on weather and visibility conditions. Therefore, runway capacity is difficult to predict and subject to large changes during operations. This often leads to congestion and large delays.

The separation required between aircraft depends on the weight categories and sequence of the aircraft involved. A light aircraft landing behind a heavy aircraft requires more separation than the reverse order. This means that the capacity can be enlarged by actively sequencing the flights. However, currently this is not done in practice.

Runway operations scheduling involves assigning a landing or take-off time and runway to each flight in such a way that the required separation between all flights is respected. By actively scheduling the flights, efficient sequences can be obtained and thus the capacity can be enlarged. This means there is an opportunity to improve the efficiency at this bottleneck and with that the efficiency of the total air traffic system.

At many airports landing and take-off operations are not combined on the same runway at the same time. The problems of scheduling landings or take-offs (at a single runway) are similar. From a practical viewpoint it can be argued that scheduling of landing aircraft is more constrained, because airborne flights have to be considered. This will lead to additional schedule restrictions, because of speed, fuel and safety considerations.

When landings and take-offs are combined at a single runway, landings usually have priority over take-offs. This is because it is easier, cheaper and safer to delay an aircraft waiting to take off in the ground than an aircraft waiting to land in the air.

In our research, the scheduling of landings at a single runway is considered first. This problem is commonly known as the (single runway) aircraft landing problem. Next, the scheduling of landings and take-offs at multiple runways is considered. The goal of the research is to assess the effects of considering airline preferences in both scheduling problems.

1.2 Research Premise

In the previous section it was identified that runway capacity can be increased by actively sequencing flights. This gives an opportunity to improve the efficiency at this bottleneck and with that the efficiency of the total air traffic system. However, efficiency cannot be considered as the only objective, because this can lead to unacceptable delays for individual flights. The enormous impact of delays on airlines and their passengers was discussed at the beginning of this chapter. Flight delay is a poor measure of this impact, because the impact of a certain delay might differ from flight to flight depending on, among others, the number of (transfer) passengers. Airlines have preferences about which of their flights should receive a (larger) delay, if necessary at all. Currently, these preferences are hardly considered in air traffic control decisions.

The goal of this research is to develop an approach to consider airline preferences in runway operations scheduling. This approach will be used to evaluate to what degree the impact of delay on airlines and their passengers can be reduced, by actively sequencing the flights at this bottleneck in the air traffic system.

A difficulty when considering individual airlines preferences is fairness. It is the role of air traffic control to assure that air traffic proceeds in a safe, efficient and equitable manner. Consequently, scarce air traffic capacity has to be assigned to competing airlines in a fair manner. Currently this is achieved by processing flights in a first come first served manner. This approach is abandoned when the sequence of the flights is optimized. Considering preferences of competing airlines also leads to fairness issues. These fairness issues will be explicitly considered and evaluated in the research.

1.3 Scope

In this section the scope of the research will be further defined.

1.3.1 Runway Operations Scheduling

In the scheduling of runway operations the sequence-dependent separation required between aircraft is considered in our research. The goal of runway operations scheduling is to determine a set of feasible landing and take-off times. These times are constrained to be feasible with respect to the required separation times between flights. Furthermore, the landing or take-off time of each aircraft is constrained to fall within a predefined time interval defined by practical considerations, such as the amount of remaining fuel for airborne flights.

Using these constraints a wide variety of different situations at different airports can be modeled in a realistic manner. This means that if a feasible schedule is obtained by the model, this schedule can be executed in reality. The model does however not provide an answer on *how* to control the flights such that the schedule will be achieved. It does, for example, not provide an answer to the question which routes landing aircraft should fly to the runway and what their speed should be.

1.3.2 Time Horizon

The planning horizon considered in the research is at most a few hours in advance (before the scheduled runway operation). In this way, accurate weather and traffic information can be used. At the same time, there is still enough flexibility to obtain considerable improvements. This allows, for example, delaying short haul flights before their departure, when considerable landing delays are expected. This means the delays are consumed on the ground instead of in the air, which is both cheaper and safer.

It is likely that disruptions occur in practice, which causes the current schedule to become infeasible. In this case rescheduling is required. Computational experiments representing such a dynamic situation are also performed.

1.3.3 Representation of Airline preferences

In this research an approach to consider airline preferences in runway operations is developed. The airline preferences are represented by cost functions. These cost functions represents the cost related to runway operations times of flights and connection times between flights. We want to allow the airlines as much flexibility as possible in representing these cost functions. At the same time, these cost functions must be applicable to establish a fair and efficient runway schedule. Therefore, it must be possible to compare the cost functions from competing airlines in a fair manner. Additionally, it should not be possible for airlines to conduct strategic behavior. To achieve this, a combination of centralized decision making and restrictions on the cost functions are proposed. Additional measures of fairness are also defined and evaluated throughout the research.

Centralized decision making is performed by an air traffic manager. The airlines are requested to communicate their cost functions to this air traffic manager. This can be done (automatically) using computer systems. This fits the current practice, where computer systems are already used to exchange information between airlines and air traffic authorities.

1.4 Overview of the thesis

In our research runway operations scheduling is considered. In Chapter 2 some background is provided on the issues related to runway operations. Runway capacity is usually the controlling element of airport capacity. The factors that that determine runway system capacity at an airport are discussed. The demand for runway operations is coming from airlines that want to offer flights from and to an airport. The supply and demand for airport capacity has to be balanced. It is explained how this is done currently and what the developments in this area are.

In Chapter 3 the existing literature about optimization models for scheduling of runway operations is discussed.

In Chapter 4 we present our approach to represent airline cost and discuss related fairness issues.

In Chapter 5 a model and a heuristic for the aircraft landing problem considering airlines' cost are introduced. Computational experiments using flight schedule data from a large European hub are also discussed. The results show that tremendous cost savings for the airlines can be obtained, especially during periods of runway congestion. Chapters 4 and chapter 5 are based on Soomer and Franx ([70],[71]) and Soomer and Koole [72].

In Chapter 6 an extension of the model is presented considering scheduling arriving and departing flights on multiple runways. This provides the possibility to use airline cost functions representing cost related to flight connections. Computational experiments were performed using these cost functions and the results are discussed. These results show that the possibility to consider the dependencies between arrivals and departures brings additional cost savings. In this way, for example, the number of missed transfers can be (further) reduced. This chapter is based on Soomer [69].

The thesis ends in Chapter 7 with a summary and conclusions.

Chapter 2

Airport Capacity and Demand

In our research runway operations scheduling is considered. In this chapter some background is provided on the issues related to runway operations.

Runway capacity is usually the controlling element of airport capacity. In Section 2.1 the factors that determine runway system capacity at an airport are discussed. The demand for runway operations is coming from airlines that want to offer flights from and to an airport. The characteristics of this demand are discussed in Section 2.2. The demand and capacity must be balanced. In Section 2.3 it is explained what is currently done to achieve this. Current and future developments in air traffic management will change these processes. The most important developments are Collaborative Decision Making and Free Flight. These developments will be discussed in Section 2.4.

All this will put forward important issues that form a motivation for our research and should be considered in runway operations scheduling. This chapter ends with a discussion of these issues in Section 2.5.

2.1 Airport Runway System Capacity

Runway capacity is normally the controlling element of the airport capacity, according to Ashford and Wright [10]. The capacity of an airport runway system can be defined in several ways and depends on prevailing conditions. A common measure is the ultimate or saturation capacity: the maximum number of aircraft that can be handled during a given period under conditions of continuous demand.

Ashford and Wright [10] group the factors that influence the capacity of an airport runway system into four classes:

1. Characteristics of demand
2. Layout and design of the runway system
3. Air Traffic Control
4. Environmental conditions in the airport vicinity

We discuss these factors in more detail.

The capacity of a single runway is determined by the separation required between aircraft. This separation is needed for safety. The required separation depends on wake vortices caused by aircraft and visibility conditions.

Every aircraft in flight trails an area of unstable air behind it known as wake turbulence or wake vortex. Trailing aircraft should avoid flying in this wake vortex. Therefore, separation requirements are stated for operations on a single runway by the International Civil Aviation Organization (ICAO). The strength of the vortex is governed by the weight and speed of the aircraft. In general, heavier aircraft fly faster. The aircraft are grouped into three weight categories (light, medium and heavy) and the separation requirements depends on the categories of the aircraft. Examples of aircraft types in the medium aircraft category are Fokker 50 and Boeing 737. A Boeing 747 falls into the heavy category. A heavier aircraft trailing a lighter aircraft needs less separation than the reverse order. The separation rules are listed in Table 2.1. Thus, runway capacity depends on the characteristics and sequence of the aircraft using the runway.

Another factor that determines the required separation distance are visibility conditions. In the U.S. visual flight rules (VFR) are used when visibility conditions are good. Under these rules pilots are responsible for maintaining separation from other aircraft. This is done by visual reference. In

		Trailing aircraft		
		Light	Medium	Heavy
	Light			
Leading aircraft	Medium	5		
	Heavy	6	5	4

Table 2.1: Wake vortex separation in nautical miles for different weight categories

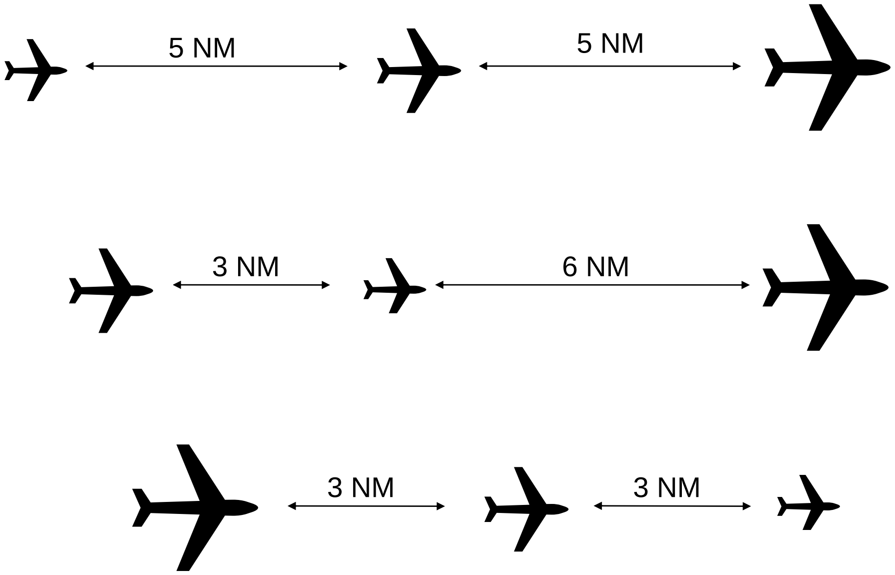


Figure 2.1: Sequence dependent separation between a light, medium and heavy aircraft.

Europe usually instrument flight rules (IFR) are used. Under these rules, air traffic control is responsible for separating aircraft. In order to achieve this, radar devices are used. The use of these devices requires a minimum separation of 2.5 or 3 nautical miles in the vicinity of an airport under good visibility conditions. Under low visibility conditions a minimum separation distance up to 9 miles might be required.

The actual required separation is determined by taking the maximum over the required wake-vortex and radar separations.

In Figure 2.1 the effect of the sequence of the aircraft on the required separation is depicted. A minimum radar separation of 3 nautical miles is assumed. The figure shows that the total required separation between one light, one medium and one heavy aircraft is between 6 and 10 nautical miles depending on the sequence of the aircraft. This leads to large differences in the time it takes to land these flights. A lot of capacity calculations are done assuming the aircraft are in random order (given the distribution over the aircraft weight categories). Actively sequencing the flights can increase capacity.

In practice, some capacity will be lost because the air traffic controller and pilots will not always be able to control the flights exactly such that the minimum required and possible separations are maintained exactly.

The total runway system capacity depends on the number of runways but also on the layout of the runway system. The operations on parallel runways (that are close together) or intersecting runways must be coordinated and the actual separation requirements depend on the exact layout and use of the runways.

On larger airports with multiple runways, there is usually a runway use program. This consists of a set of (preferential) runway configurations that can be used. Such a runway configuration is a combination of runways and their use (the types and direction of operations). Runways can be used in segregated mode or mixed mode. Segregated mode means arriving aircraft are allocated to one runway and departing aircraft to another. In mixed mode landings and take-offs are combined at the same runway. There exist specific separation rules for a landing followed by a take-off and a take-off followed by a landing.

The active runway configuration at a certain time is determined by Air Traffic Control and depends on the environmental conditions and the demand at the time. It is convenient to use runways that are nearly aligned with the wind (for head wind operations). Noise regulations can also be a consideration in choosing a runway configuration.

To illustrate the above, the use of the runway system at Amsterdam Airport Schiphol is described. This information is provided on the website of the Dutch air traffic control organization LVNL [53]. Amsterdam Airport Schiphol has 5 runways for commercial airlines. These runways are usually used in segregated mode. During peak periods three runways are used at the same time. Depending on the traffic supply these are combinations of two runways used for take-offs and one for landings or one for take-offs and two for landings. The total of 5 runways gives multiple combinations of three runways. Weather conditions and noise regulations determine which combination will be used. In peak hours more than 100 flights per hour are handled.

As illustrated above, several factors have an impact on runway (system) capacity. This makes it complicated to define runway capacity in an unambiguous manner and leads to a number of different approaches that may be employed to estimate runway (system) capacity. These include analytical approaches, such as the classic paper of Blumstein [22] and the use of queuing models, see for example the papers of Bäuerle et al. [16] and Stolletz [74]. These approaches show that when demand is close to capacity delays to aircraft increase sharply.

2.2 Demand for Airport Capacity

The demand for airport capacity comes from airlines that want to offer flights from and to the airport based on expected customer demand. The design of an airline flight schedule containing flights at desirable times in profitable markets is a complex process. This process typically begins more than 12 months prior to operation. According to Lohatepanont and Barnhart [52] schedule design has traditionally been decomposed into two sequential steps:

1. Frequency planning: The appropriate service frequency for each origin-destination pair is determined.
2. Timetable development: The proposed services are placed throughout the day subject to network considerations and other constraints.

The flights of the airlines are usually not evenly distributed over the day. A lot of (business) travelers want to travel early in the morning and at the beginning of the evening, which leads to a larger number of flights at these times.

To make efficient use of their resources (aircraft and crew) many (large) airlines use a so-called hub and spoke network. This entails consolidating traffic from a diverse range of origins to a diverse range of final destinations at large hub airports. This network structure has large implications for the daily spread of flights. Hub airlines schedule subsequent series of arrivals and departures to maximize customer choice (transfer possibilities)and minimize customer travel times. These series are called banks. This leads to higher load factors and thus increases revenues. Button [25] argues that the efficient resource use and load factors usually outweigh the disadvantages such as higher operational cost of ground handling caused by the concentration of the operations.

On large hubs it is not uncommon to have more than five banks a day. During these periods, the demand of the airport is close to the capacity or even sometimes exceeds this capacity (e.g. under low visibility conditions).

This results in congestion. Recently, some airlines are considering (some level of) depeaking of their schedules to limit the amount of congestion. This is done by carefully weighing customer service and operational costs. Lufthansa obtained positive results by depeaking their schedule at their hub Frankfurt International Airport [39]. With the new schedule structure, the overall travel time for 35 of the 50 most profitable flight connections was decreased. Moreover, ground delays went down by about 50% and flight times inbound Frankfurt decreased due to reduced congestion.

2.3 Balancing Demand and Capacity

The amount of flights and passengers have been increasing for decades. After an interruption caused by the 2001 September 11 attacks, this trend continues. The European air traffic management organization, EUROCONTROL, witnessed in 2007 an acceleration in the rate of growth of daily air traffic within the European region with an increase of 5.3% compared to 2006 [38]. There was a 23% increase in daily air traffic compared to 2002. This growth is not being met by corresponding increase in the physical capacity of the system (such as new airports and runways).

To balance runway demand and capacity, different processes exist at the strategic, tactical and operational level. At some airports airlines must have been allocated arrival and departure slots in advance in order to operate. Slot coordination is discussed in Section 2.3.1. Both at a strategical and tactical level, air traffic flow management is performed to balance en-route and airport capacity (on a relatively aggregate level) by adjusting routes, departure times and speeds of aircraft. This is discussed in Section 2.3.2. Air Traffic Control (ATC) is responsible for guiding aircraft during operations and thus handle congestion and conflicts. ATC procedures related to runway operations are discussed in Section 2.3.3. In Section 2.3.4 it is explained how airlines handle operational disruptions that are (amongst others) caused by air traffic congestion.

2.3.1 Slot Coordination

The International Air Transport Association (IATA) makes the following distinction in levels of airport capacity as published in the Worldwide Scheduling Guidelines (WSG) [44]:

Level 1: Airports with adequate capacity to meet demands (Non-coordinated airports)

Level 2: Airports where demand is approaching capacity (Schedules facilitated airports)

Level 3: Airports where demand exceeds capacity and no solution to be expected in the short term (Coordinated airports)

At coordinated airports (Level 3), airlines must have been allocated a slot before operating. The IATA defines a slot in the following way : "a scheduled time of arrival or departure available for allocation by, or as allocated by, a coordinator for an aircraft movement on a specific date at a coordinated airport". The coordinator should be appointed by the appropriate authority and its activities must at all times be neutral, transparent and non-discriminatory. Guidelines for the process of coordination are also provided by the IATA in the WSG.

To facilitate the process, the IATA organizes schedule coordination conferences in which airlines participate. These are organized twice a year, about four months before the start of the summer and winter scheduling seasons that are to be discussed. Airport capacity limitations applicable for the season under discussion are declared before the conferences by the appropriate authorities in consultation with airlines. About three weeks before each conference, airlines provide coordinators with slot requests for the arrival and departure times required at the airports concerned. The coordinator collates this information and identifies periods in which slot requests exceed declared airport capacities. Slots are allocated based on historical precedence. An airline using a slot during one season keeps it for the following season as long as the slot has been properly used. Of the remaining slots at least 50% must be allocated to new entrants (if there are enough requests). During the conference, schedules are adjusted mainly through bilateral discussions between airlines and coordinators regarding alternatives offered, or between airlines to exchange slots offered or accepted.

In Europe a few dozen airports (most of the large hub airports) are slot coordinated. In the U.S. currently only three airports, Chicago O' Hare, Newark and New York J.F.K., are slot coordinated.

Slot coordination is a strategical process. Weather conditions, delays and other operational perturbations can still cause airport demand to (temporary) exceed capacity during operations.

The slot allocation is currently an administrative process and the slots are free of charge. This process does not provide incentives to request slots outside peak periods. Especially hub airlines will request many slots in peak periods, which will often leads to congestion and delays in these periods. Daniel and Harback [31] show that this negatively affects other airlines at many large hubs in the U.S. Alternative slot allocation methods to overcome this problem are often proposed in the literature: Congestion pricing involves charging airlines for slots in peak periods, see for example Daniel [29], [30] and Johnson and Savage [47]. Slot auctions are also proposed, see for example Rassenti et al. [60], Loan et al. [51] and Ball et al. [12].

Schank [66] observes that market based slot allocation methods have been scarcely (successfully) applied in practice: At London Heathrow, Boston Logan and Chicago 'O Hare attempts to implement peak runway pricing were made, but these were not effective. This is attributed to difficulties with the application of free market principles in a government-regulated market. These institutional barriers and the lack of adequate substitutes for air travel have hampered the successful application.

2.3.2 Air Traffic Flow Management

To balance en-route and airport demand and capacity, keeping delays to a minimum and avoiding congestions and overload, flow management is performed in some (busy) areas. Flow management is performed on a relatively aggregated level, both in time and area considered. A flight will usually be subjected to flow management prior to being handled by Air Traffic Control during operations. A broad overview of air traffic flow management, both from a practical and theoretical perspective is given by Hoffman et al. [43].

In Europe flow control is performed permanently in a large part of the European airspace by the Central Flow Management Unit (CFMU) of EUROCONTROL. The information used in this section is compiled from their website [37]. In the U.S. the Federal Aviation Administration (FAA) is responsible for air traffic management. There, air traffic flow management is not used frequently to balance en-route capacity and demand. However, ground delay programs are commonly used to balance airport (arrival) capacity and demand. Ground delay programs are discussed below.

Strategical flow management consists of preparing routing schemes that balance the expected air traffic flows in order to ensure maximum use of airspace and minimize delays. This is done by the CFMU during the period from several months until a few days before a flight.

Based on traffic forecasts a tactical plan for the next day is prepared and airlines and air traffic control will be informed about flow management measures that will be in force on the following day.

On the day of operations, before the flight has departed, flow management can propose alternative departure times, re-routings to avoid bottlenecks and alternative flight profiles to maximize efficiency. This is done in collaboration with the airlines.

According to the CFMU the overwhelming majority of flights are not subject to flow restrictions. Flow management has made a significant contribution to en-route throughput enhancement and delay management.

Ground Delay Programs

An important part of flow management decisions consists of so-called ground delay programs. A Ground Delay Program (GDP) is issued when it is expected that airport demand will exceed capacity for a sustained period of time. Usually this is caused by a decline in airport arrival capacity (e.g., because of severe weather conditions). To balance demand and capacity, delays have to be assigned to flights planning to land at the airport in the considered period. During a GDP these delays are (as much as possible) assigned to flights on the ground at their origin prior to departure rather than en-route. According to Inniss and Ball [45] it is about twice as cheap to delay an aircraft on the ground instead of in the air. Next to this, it is safer to delay flights on the ground than in the air.

The following steps are performed when a GDP is issued. The (expected) airport arrival capacity during the period of congestion is estimated. A number of landing slots equal to this capacity are defined. These landing slots are allocated to flights that were originally scheduled during the period of congestion. By subtracting the flight times from the assigned landing times, updated departure times are determined for the flights. Thus, ground delays at the origin are assigned to these flights.

The slots assigned in a ground delay program do not necessarily correspond to slots considered in slot coordination: Airports that are not slot coordinated can still be subject to a ground delay program.

Although, in the U.S. air traffic flow management is not used in many areas to balance en-route capacity and demand, ground delay programs are

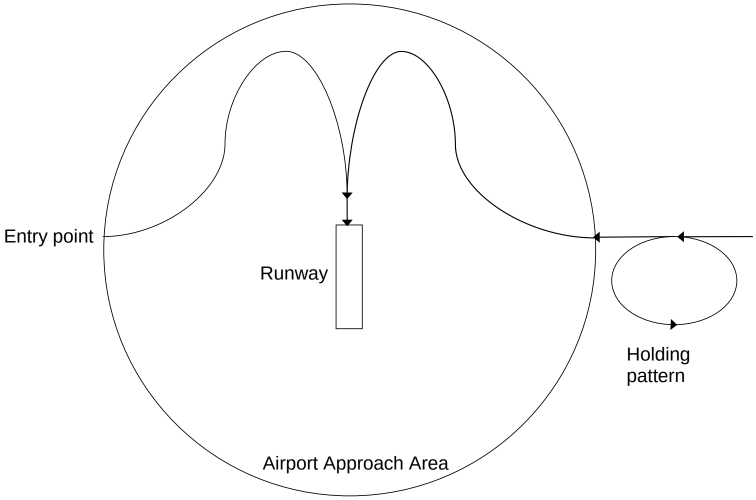


Figure 2.2: Schematic overview of the Airport Approach Area

commonly used to balance airport (arrival) capacity and demand. According to Chang et al. [27] anywhere from 600 to 1,000 ground delay programs are enacted in the U.S. each year.

A more detailed literature overview of (mathematical modeling of) ground delay programs is given in Section 3.2.

2.3.3 Air Traffic Control

Air Traffic Control (ATC) is responsible for guiding aircraft in a safe, equitable and efficient manner. This also involves the balancing of demand and capacity during operations. This is especially important around (busy) airports.

Flights approaching the airport are under the guidance of the so-called approach controller from approximately 50 miles from the runway, when they enter the airport approach area. From this moment on, the controller must create a correctly separated flow of aircraft towards the runway(s). Because of the limited time available and the high workload of the controllers, hardly any changes to the sequence of the flights can be made at this point.

In Figure 2.2 a schematic overview of the airport approach area is shown. There are a limited number of entry points to the approach area. From each entry point there is a route (Standard Terminal Arrival Route) to the runway.

The last part of these routes is the common approach path to the runway. On all these routes separation between aircraft has to be maintained. In order to create a correctly separated flow of aircraft, the controller can order speed changes. In practice the approach controller gives radar vectors and altitude restrictions to separate aircraft inside the approach area and to guide aircraft to the final approach path. In case of congestion, the controller can temporarily prevent aircraft from entering the approach area by ordering aircraft (possibly several times) to fly a holding pattern. This is a fixed diversion leading to the same point again.

The approach controller also monitors departing aircraft and assures separation from other traffic. A series of Standard Instrument Departure (SID) routes starting at the runway are defined for departing aircraft, analogue with the Standard Terminal Arrival routes used for arrivals.

2.3.4 Airline Operations Control

Air Traffic Flow Management and Air Traffic Control measures have impact on airlines' operations. Arrival delay may cause departure delay for subsequent flights using the same aircraft or crew. In this way the delay will propagate. In order to address issues occurring during operations, airlines have an operations control center. This center will try to recover the operational plans related to crews and aircraft assignments and passenger connections. This is done by reassigning crew and aircraft to different flights, but it might also be necessary to delay or cancel flights.

It is complicated to oversee all the implications of possible recovery measures and to select the optimal action. Therefore, it is natural to use mathematical models to achieve this. This subject receives a lot of attention in the scientific literature: Examples are the papers of Lettovsky et al. [50] and Stojkovic et al [73], both considering crew rescheduling. Bratu and Barnhart [23] consider aircraft, crew and passenger recovery. Next to this, the (initial) schedule, crew and aircraft assignments can be designed in such a way that they are robust with respect to operational disruptions. See for example the papers of Lan et al. [49] about robust schedule design, Smith and Johnson [68] about robust aircraft assignment and Shebalov and Klabjan [67] about robust crew assignment.

It is clear that a delay at the day of operation of a flight (because of congestion in the air traffic system) will incur costs for the airline. A study performed by the Transport Studies Group of the University of Westminster and published by EUROCONTROL [28] estimated these costs. The results show that the cost *per minute delay* increase with the total time of delay and the number of (occupied) seats of the flight. The average cost per minute of a short delay (around 15 minutes) range from below 1 euro to almost 50 euro depending on the number of (occupied) seats. The average cost per minute of a long delay (around 65 minutes) range from around 30 euro to almost 300 euro. These costs include the network effects of the delay (schedule recovery and the resulting delays and cancellations of other flights).

Beatty et al. [20] introduce the concept of a delay multiplier to estimate the total system impact of a delayed flight. A delay multiplier is applied to the initial delay of an aircraft to estimate the amount of propagated delays to all flights connected to the initial flight by crew or by aircraft. Large delays early in the day are most disruptive. The delay multiplier grows nonlinearly with the length of the initial delay.

From interviews with experts from a major European Airline, it became clear that their cost caused by arrival delays for flights at the hub airport are for a large part related to transfer passengers missing their connections.

2.4 Developments

In the previous section air traffic management processes were discussed. Current and future developments in air traffic management will have a large impact on these processes. The most important developments are Collaborative Decision Making and Free Flight, which will be discussed in this section.

2.4.1 Collaborative Decision Making

As stated in Chapter 1, aggregate efficiency metrics used by air traffic managers do not represent the impact of delays on individual airlines very well. Collaborative Decision Making (CDM) is an effort to improve air traffic management through information exchange, procedural improvements, tool development and common situational awareness between air traffic managers and airlines. Both EUROCONTROL and the FAA have programs to incorporate CDM in several processes. See for example the European CDM portal [36]. A detailed overview of CDM in air traffic flow management is given by Hoffman et al. [43].

A basic step in applying CDM is (automated) information exchange between Airline Operations Control Centers, Air Traffic Flow Management and Air Traffic Control organizations. Earlier notification of flight cancellations and congested airspace or airports can improve the decisions of all parties. The (timely) availability of information to all parties, could also lead to partly shift the decision responsibility from air traffic managers to airlines.

Andersson et.al [6] performed a simulation study to assess the benefit from increased communication (real time exchange of data) and collaboration during the arrival process. The results are based on a simulation study based on historical arrival data. The data shows that the difference between the (airline) expected landing time and the realized landing time is on average 5 minutes. The simulation shows that decreasing the standard deviation of the landing time estimate error from 5 minutes to 3 minutes could avoid 500 passenger minutes of delay during a 3-hour period during normal operations. Doing so during a busy, heavily-delayed 3-hour period could avoid 2,000 passenger minutes of delay, which is a reduction of 3%. For the same time periods, allowing an airline to influence the sequence of their arriving flights, saves 4,000 passenger minutes of delay in the on-time day period and 7,000 passenger minutes of delay in the busy day period.

CDM in Ground Delay Programs

The most advanced implementation of CDM has been for Ground Delay Programs (see Section 2.3.2) in the U.S., see Chang et al. [27].

The program did not only include the exchange of information but also included procedural changes. As Vossen and Ball [76] discuss, prior to the implementation of CDM in 1998, the GDP arrival slots were allocated to flights in a First Come, First Served manner based on the most recent estimated arrival times. This gave the airlines a disincentive to report flight delays, because this would result in a later estimated arrival time. That would likely result in an additional delay because the flight would be assigned to a later arrival slot. To avoid this, airlines did not report delays (in a timely manner) and the GDP decisions were based on poor data and were therefore often inefficient.

Since the implementation of CDM in 1998, GDP slots are allocated to *airlines* instead of flights. The airline can decide which of the slots assigned to it to use for which of its flights. The initial allocation of the slots to airlines is based on the original flight schedule, as opposed to the most recent estimated schedule. If an airline is not able to use one of its slots because of delays or cancellations, it will receive another slot (as early as possible) in return, if possible. This method provides incentives for airlines to report delays and cancellations as early as possible.

Ball et al. [14] evaluated the results. Participating airlines send operational schedules and changes to schedules to the Air Traffic Control Systems Command Center (ATCSCC) on a continual basis. This schedule information includes, but is not limited to, flight delay information, cancellations, and newly created flights. Through the use of the Flight Schedule Monitor (FSM), the ATCSCC uses this information to monitor airport arrival demand and to conduct ground delay programs. The airlines are also able to monitor arrival demands and model ground delay programs via FSM but do not have the capability to alter or implement ground delay programs. The results show an increase in the accuracy of departure time estimates. Flight cancellations are also known earlier to all parties (especially the ATCSCC). In the past cancellations were known after the original departure time, but now they are known well before the original departure time. Based on a series of interviews, a consensus among ATCSCC specialists is that CDM procedures yield more effective ground delay programs. Data also show a decrease in the total minutes of (departure) delays assigned during these programs.

A more detailed literature overview of (mathematical modeling of) ground delay programs under CDM is given in Section 3.2.

2.4.2 Free Flight

The concept commonly known as Free Flight represents a paradigm shift for the air traffic management system. It entails the shift of operational responsibility (for maintaining separation) from air traffic control to the pilot. Nowadays, the concept is also referred to as *airborne self-separation operations* or *autonomous aircraft operations*. The Next Generation Air Transportation (NextGen) Institute [46] is a combination of six U.S. government agencies that is devoted to develop the concept.

The concept is enabled by advances in technology. A system that warns the pilot of possible conflicts is required for safety reasons. Such a system is known as a Airborne Separation Assurance System (ASAS), see Hoekstra [41]. In the past it was not possible to incorporate such a system in a cockpit and therefore air traffic control is (still) performed centrally from the ground. In order to do this efficiently, only designated areas (routes) are used for air traffic, currently. However, this restriction is not needed when pilots are responsible for maintaining separation. Abandoning this restriction leads to a larger geographical dispersion of air traffic and thus to a lower probability of (potential) conflicts. Another big advantage of the concept is that airlines can select the most efficient routes with respect to fuel use.

Post and de Jonge [57] remark that (large hub) airports will remain a bottleneck in the system under the free flight concept. Landings and take-offs at these airports will still require air traffic management and control. A regular, balanced supply of arrival traffic is essential for successful planning on dense arrival flows.

Andreatta et al. [8] state also that arrival planning plays an important role in the free flight concept: The concept was born among the air carriers of the United States who were asking the FAA to provide them only with the arrival time slot, leaving the airlines the freedom of selecting the departure times, routes and speed of their aircraft.

2.5 Discussion

The volume of air traffic is increasing fast and is not being met by corresponding increases in the physical capacity of the system (such as new airports and runways). Flow control has achieved significant improvements in enlarging the en-route traffic throughput. As a result, the air traffic bottleneck is shifting from the en-route segments to the airports. Under the free flight paradigm this will be even stronger and runway operations scheduling will still play an important role.

The demand at large hub airports is esubject to large peaks, because of the hub and spoke network that most airlines use. Airport runway capacity is unpredictable and subject to large changes during operations, mainly because of weather and visibility conditions. This often leads to imbalance between capacity and demand. Several processes exist to handle this imbalance. On a strategical level slot coordination is used at some airports. Tactically (on the day of operations), flow management control measures such as ground delay programs can be issued. Air Traffic Control is responsible to address congestion and conflicts during operations.

Runway capacity depends on the weight categories of the aircraft and their (landing and take-off) sequence. However, in practice not much is done to actively sequence flights. The approach controller has little time left to adjust the sequence of landing flights.

The above reasons confirm the importance of (tactical) runway operations scheduling in order to use scarce runway capacity in an efficient manner. But, focusing only on efficient sequences can result in unacceptable delays for individual flights. This could have a large impact on passengers and airline processes, e.g., missed transfers and crew or aircraft scheduling problems. However, flight delay is a poor measure of the impact for airlines and passengers, because the impact of a certain delay might differ from flight to flight depending on, amongst others, the number of (transfer) passengers.

This leads to the idea of considering airline preferences for individual flights when scheduling runway operations. This fits in with the current focus on Collaborative Decision Making. By doing this, it is expected that the impact of delays for airlines and their passengers can be reduced and the airlines need less recovery measures. Currently, the cost of delays and related recovery measures are considerable.

Chapter 3

Literature Overview

The problem of scheduling runway operations is, because of its complexity, extremely suitable to be analyzed using mathematical models. The subject has indeed received considerable attention in operations research literature. In this chapter, this literature will be discussed.

Two types of mathematical models related to the runway scheduling of (landing) flights exists. The Aircraft Landing Problem (ALP) focuses on capacity by optimizing the landing sequence of the flights given the (sequence dependent) separation that is required between flights. Literature considering the ALP is discussed in Section 3.1.

A Ground Delay Program (GDP) is used in a sustained period of congested landing runways. Available capacity for the considered period is given as input. The problem focuses on the assignment of this capacity in a cost effective and fair manner to flights. Literature considering GDPs is discussed in Section 3.2.

In Section 3.3 the contributions of this research compared to the literature are discussed.

3.1 Aircraft Landing Problem

The aircraft landing problem considers the scheduling of aircraft landings at a single runway or multiple runways at an airport. The problem involves determining a landing sequence for the set of aircraft and a landing time for each aircraft, respecting the required separations between the flights. The quality of schedules is evaluated using an objective function, e.g., total delay. The resulting schedule can be compared to the First Come First Served

(FCFS) schedule. In the latter schedule it is assumed that the flights are landed (respecting the required separation) in the order in which they approach the airport. This is similar to the way flights are handled in practice currently.

The optimization models found in the literature model various characteristics of the problem differently. We will first discuss these characteristics (Section 3.1.1) and the overall complexity (Section 3.1.2) of the problem. Next, the models are discussed, grouped by the solution techniques, in more detail. This will be done first for the single runway problem in Section 3.1.3 and then for the multiple runway problem in Section 3.1.4. In Section 3.1.5 a general discussion of the ALP literature is given.

Notation

Throughout this thesis the following notation will be used to represent the ALP:

Let $F = \{1, \ldots, N\}$ be the set of flights to schedule.
Let

E_i :	Earliest possible landing time for flight i	$i \in F$
L_i :	Latest possible landing time for flight i	$i \in F$
S_{ij} :	Required separation time when flight i lands before flight j at the same runway	$i, j \in F, i \neq j$

The problem is to determine a landing sequence and landing times t_i for each flight $i \in F$, such that $t_i \in [E_i, L_i]$ and $t_j \geq t_i + S_{ij}$ when flight i lands (immediately) before flight j.

3.1.1 Problem Characteristics

Separation Times

As explained in Section 2.1, the required minimum separation *distances* are based on the weight categories of the aircraft and are sequence dependent. In [5] and [59] it is assumed that the required separation *times* are determined just by these weight categories. In practice, the situation is usually more complex, because (different) separation distances have to be maintained at different points of the approach area and because the approach speeds of the aircraft differ. This can be modeled by allowing for a custom separation time between every pair of aircraft. Separation can be considered only between successive landings (successive separation) or between all pairs of flights (complete separation). These cases are equivalent if the triangle inequality holds for all separation times: $S_{ik} \leq S_{ij} + S_{jk}$ for all $i, j, k \in F$. Successive separation is assumed in [21], [32], [58] and [59]. Complete separation in [4], [17], [18], [19], [34], [35] and [56],

Landing Time Interval

It is common to consider the landing time for each flight $i \in F$ to be constrained to a landing time interval $[E_i, L_i]$. Such an interval is considered in [4], [5], [9], [17], [18], [19], [34] and [56],.

However, in [32], [58] and [59] it is assumed all flights considered are currently waiting to land, which is equivalent to $E_i = 0$ for each flight.

In [21], [32], [35], [58] and [59], it is assumed that the latest feasible landing time is sufficiently large to be of no consequence ($L_i = \infty$). To limit the amount of delay under this assumption, [32], [58] and [59] constrain the number of positions a flight can be shifted (forward or backward) compared to the FCFS sequence.

Runway Assignment

When multiple runways are considered each flight has to be assigned to land on one of the runways. This is considered in [17], [18], [34], [56] and [58]. Multiple runway problems are discussed in Section 3.1.4.

Objective

Different objectives are considered in the literature. The two most commonly used objectives are to maximize the total throughput and to minimize the total delay.

Total throughput is used as the objective when the focus is on efficient use of runway capacity and therefore preferable from an airport and air traffic control perspective. The goal is to minimize the total time required to land all flights or equivalently to minimize the latest landing time:

$$\min \max_{i \in F} \{t_i\}.$$

This objective is considered in [5], [21],[58] and [59].

A more airline oriented objective is to minimize the total delay (compared to the earliest possible landing time):

$$\min \sum_{i \in F} (t_i - E_i).$$

Weights can be used to express relative importance of individual flights. This objective is considered in [5], [19], [21], [58] and [59].

Using either of these objectives, it is straightforward to determine optimal landing times *for a given order of the flights*. In this case, it is optimal to land every flight as early as possible: Suppose the flights land in the order of their indices i, than the optimal landing time of the first flight $t_1 := E_1$ and for successive flights $t_i := \max\{E_i, t_{i-1} + S_{i-1,i}\}$.

Another airline oriented objective is to consider the deviation from a target landing time $\hat{t}_i$ for each flight. The objective becomes

$$\min \sum_{i \in F} |t_i - \hat{t}_i|.$$

Different weights can be used for different flights and for deviations before target time and deviations after target time. With this type of objective, it becomes more complicated to determine the optimal landing times for a given sequence. In [4], [17], [18], [19], [34] and [56] deviation from a target time is considered as objective.

Note that the latter two objectives do not distinguish between solutions with the same total delay (or deviation). This delay can however be divided over a large number of flights, each receiving a small delays or over a small number of flights, each receiving a large delay. The latter solution will affect some flights and airlines disproportionally and thus raises questions related to fairness.

Static and Dynamic Scheduling

The problem can be considered in a static or a dynamic manner. In the latter case, new flights appear and have to be scheduled. Therefore the current schedule has to be revised. This involves freezing some part of the current schedule considering the past and near future. It is usually not wanted to deviate too much from the current schedule, anyway. Therefore, the current schedule is sometimes used as reference schedule. In the objective, deviations from this reference schedule are then minimized.

Dynamic scheduling also allows for considering operational disruptions that occurred and possibly caused the earlier schedule to become infeasible.

Dynamic scheduling is considered in [18], [32] and [35].

3.1.2 Problem Complexity

The problem is equivalent to a job shop scheduling problem. The jobs are the aircraft to land and the runways represent the machines. The separation times between consecutive pairs of flights are the (sequence dependent) processing times. The earliest possible landing time represents the release time of the job and the latest the due date. A common objective is to minimize the completion time (landing time) of the latest job, which is equivalent to maximizing throughput.

The single machine problem is equivalent to the single runway problem. The single machine problem with equal release times and no due dates was proven to be NP-hard by Rinnooy Kan [64]. The problem with unequal release times but processing times that are not sequence dependent has been proven to be strongly NP-hard by Brucker et al. [24]. It follows that the problem with sequence dependent processing times is also strongly NP-hard. This implies that no efficient (polynomial time) algorithm exists to solve these types of problem optimally. Practically, this means that for larger instances heuristics are required to find good solutions within reasonable computation time. In the next section different solution techniques (optimal algorithms as well as heuristics) used in the literature for the ALP are discussed.

3.1.3 Single Runway Problems

The single runway problem is the problem that is considered most often in the literature. In this section we will discuss the different solution techniques for this type of model.

Enumeration approaches

To find an optimal solution, complete enumeration of all possible sequences can be used. This requires to evaluate $N!$ sequences and is only practically feasible for very small values of N.

However, the number of sequences can be limited by applying a rolling horizon approach. This approach starts by evaluating all sequences of the first $n < N$ aircraft, selects the best sequence and fix the first aircraft in that sequence. Then aircraft $n + 1$ is added to the problem and the process is repeated.

Another way to limit the number of sequences is to constrain the sequences that are considered. Dear and Sherif [32] proposed an algorithm in which only sequences in which no flight is assigned to land more than a prespecified number of positions (forward or backward) from its FCFS position are considered. This algorithm is called Constrained Position Shifting (CPS). It considers the problem on a short time horizon to sequence the flights in the airport approach area. The maximum position shift reflects the difficulty for controllers to achieve large position shifts on a short time horizon in practice. Latest possible landing times L_i are not explicitly considered, but are in a way bounded by using a (small) maximum position shift. Successive separation between aircraft is enforced. A dynamic approach is considered, in the sense that the schedule is revised when a new flight enters the approach area. This is similar to using a rolling horizon approach.

Simulation experiments where demand is close to capacity, show significant average delay reductions compared to the FCFS schedule, ranging from 16% to over 60%. These experiments were performed with a maximum position shift of 4 positions. The primary objective was to maximize throughput. Next, the maximum throughput solution with minimum delay was selected. The algorithm is however only practically useful when scheduling a small number of flights using a small maximum position shift. Different objectives can be considered using this approach.

Dynamic Programming

Dynamic Programming can be used to solve problems with a recursive structure. The problems can be solved more efficiently compared to complete enumeration, by exploiting this structure.

Psaraftis [59] proposes a dynamic programming approach that gives an optimal solution for the static aircraft landing problem with category based separation times and without landing time intervals. The separation times are determined solely by the weight categories of the aircraft. Thus, aircraft are identified only by their weight category. In this way, the state space can be limited to the number of light, medium and heavy aircraft that still have to be scheduled, denoted by n^L, n^M and n^H respectively, combined with the type $T \in \{L, M, H\}$ of the aircraft that has just landed.

Let $f(T_i, T_j, n^L, n^M, n^H)$ be the (prescribed) cost of landing a category T_j aircraft immediately after a category T_i aircraft given the numbers of aircraft left. The goal is to identify a sequence of aircraft types $T_1, \ldots, T_N$, with $T_i \in \{L, M, H\}$ that minimizes the total cost. By choosing $f(x, y, n^L, n^M, n^H) = S_{xy}$ the throughput is considered by (repeatedly) adding the separation time required for the flight added. The total delay is considered by choosing $f(x, y, n^L, n^M, n^H) = (n^L + n^M + n^H)S_{xy}$. Here, every flight that is not yet scheduled will receive an additional delay that is equal to the separation time required for the flight just added.

The value function $V(T, n^L, n^M, n^H)$ is defined as the minimum total cost to land the remaining aircraft, given that an aircraft of type T has just landed. The dynamic programming recursion is given by:

$$V(T, n^L, n^M, n^H) = \min_{x \in X} \left[f(T, x, n^L, n^M, n^H) + V(x, \hat{n}^L, \hat{n}^M, \hat{n}^H) \right]$$

where $X = \{x : n^x > 0\}$ and

$$\hat{n}^T = \begin{cases} n^T - 1 & \text{if } T = x \\ n^T & \text{otherwise} \end{cases}$$

The problem can be solved by starting from $V(T, 0, 0, 0) = 0$, $\forall T$. Let N^L, N^M, N^H be the total number of light, medium and heavy aircraft, respectively, that have to be scheduled and $N = N^L + N^M + N^H$. The problem (with 3 weight categories) can be solved in $O(N^3)$ time.

The dynamic programming formulation is also modified to incorporate constrained position shifting. This makes the computation more efficient than the algorithm of Dear and Sherif [32] and thus allows for a larger maximum position shift (MPS).

In order to do this, the following notation is introduced: Let x_i be the weight class of the aircraft at the i-th position in the FCFS sequence. Given a state (T, x, n^L, n^M, n^H), the position of the current aircraft is given by $N - (n^L + n^M + n^H)$. The initial FCFS position of this job can be found by scanning the FCFS sequence from x_1 to x_N until $N^T - n^T$ aircraft of class T are encountered, say at position $p(T, N^T - n^T)$. The position shift of this aircraft is thus $p(T, N^T - n^T) - N + (n^L + n^M + n^H)$. Only sequences where the position shift is smaller than the maximum position shift (MPS) are allowed, This can be incorporated in the dynamic programming recursion by setting $V(T, n^L, n^M, n^H) = \infty$, if $|p(T, N^T - n^T) - N + (n^L + n^M + n^H)| >$ MPS.

Because of the category based approach and the absence of landing time intervals, the practical relevance of this model is limited. Landing times are not explicitly assumed, since it is always assumed flights land as early as possible.

Bianco et al. [21] give a dynamic programming formulation that includes earliest landing times (but no latest landing times) and custom successive separation times between every pair of aircraft. The objective is to minimize total delay. The state space (G, t, j) is defined by the set of flights already scheduled G, with flight j as latest scheduled flight at time t. The value function represents the minimum sum of completion times given the current state. The dynamic programming recursion is given by:

$$V(G,t,j) = \min_{(G',t',i)\in\Gamma^{-1}(G,t,j)} \Big[V(G',t',i) + (N - |G| + 1)S_{ij} + (N - |G|)\max\{0, E_j - t' - S_{ij}\}\Big]$$

with $\Gamma^{-1}(G, t, j)$ the set of states that can reach state (G, t, j):

$$\Gamma^{-1}(G,t,j) = \begin{cases} \{(G \setminus \{j\}, t', i) : t \geq t' + S_{ij}\} & \text{if } t = E_j \\ \{(G \setminus \{j\}, t', i) : t = t' + S_{ij}\} & \text{if } t > E_j \end{cases}$$

It is proved that the following general state dominance rule holds: State (G, s, j) is dominated by state (G, t, j) if $V(G, t, j) \leq V(G, s, j)$ and $s \leq t$. Note that the size of the state space becomes very large, for large N. This makes it practically impossible to use this dynamic program to solve larger instances of the problem.

Mathematical Programming

In a Mathematical Programming formulation, both the objective functions and the constraints are formulated in terms of a set of decision variables. The constraints are usually formulated as inequalities. In Linear Programming (LP) both the objective function and the constraints are formulated in linear terms of the decision variables. Linear Programs can be solved polynomially with respect to the number of (continuous) decision variables and constraints. Standard algorithms to solve a LP, such as the simplex algorithm, exist.

A Mixed Integer Program (MIP) is an LP in which a subset of the decision variables is restricted to integer values. Generally speaking, MIP problems are not polynomially solvable.

Abela et al. [4], Beasley et al. [17] and Ernst et al. [34] formulated similar MIP formulations of the static problem with complete separation and landing time intervals. The objective considered is a weighted deviation from target landing times.

Let $\hat{t}_i$ be the target landing time of flight i. Let A_i (B_i) be the time unit penalty for landing earlier (later) than $\hat{t}_i$.

The main decision variables are the landing times of the flights. The formulation requires some additional decision variables.

$$\begin{array}{lll}
t_i & : \text{landing time for flight } i & i \in F \\
\alpha_i & : \text{time units that flight } i \text{ lands before } \hat{t}_i & i \in F \\
\beta_i & : \text{time units that flight } i \text{ lands after } \hat{t}_i & i \in F \\
\delta_{ij} & = \begin{cases} 1 & \text{if flight } i \text{ lands before flight } j \\ 0 & \text{otherwise} \end{cases} & i, j \in F, i \neq j
\end{array}$$

The objective is given by:

$$\min \sum_{i \in F} (A_i \alpha_i + B_i \beta_i)$$

with the following constraints to set the time deviations:

$$\begin{array}{rcll}
\alpha_i & \geq & \hat{t}_i - t_i & i \in F \\
\beta_i & \geq & t_i - \hat{t}_i & i \in F \\
t_i & = & \hat{t}_i - \alpha_i + \beta_i & i \in F \\
\alpha_i, \beta_i & \geq & 0 & i \in F
\end{array}$$

The next constraint ensures that the landing time falls in the landing time interval:

$$E_i \leq t_i \leq L_i \qquad i \in F$$

The following constraint is used to enforce the sequence of the flights

$$\delta_{ij} + \delta_{ji} = 1 \qquad i, j \in F, j > i$$

And the separation between the flights can be assured using the following constraint:

$$t_j \geq t_i + S_{ij} - \delta_{ji} M \qquad i, j \in F, j \neq i$$

where M is a sufficiently large positive constant.

Note that, for a given order of the flights all the integer variables (δ_{ij}) are fixed. This means that the problem of finding the optimal landing times for a given order of the flights can be formulated as an LP problem, which can be solved efficiently.

The standard algorithm to solve a MIP problem is called the Branch and Bound algorithm. This algorithm searches the solution space using a binary search tree. In every node of the tree an LP problem is solved. The LP problem considered in the root node is the MIP problem without the integrality constraints. If the solution to this problems gives integer values for the (originally) integer constrained variables, the optimal solution of the MIP is found. If not, one of these variables is selected as so-called branching variable. In the above MIP, we would select one of the δ_{ij} variables for which $0 < \delta_{ij} < 1$ in the current solution. Two new problems are added as branches of the current node of the tree. These problems each have an additional constraint, either $\delta_{ij} = 0$ or $\delta_{ij} = 1$ for the current branching variable δ_{ij}. Branches can be discarded if the problem is infeasible or the objective value is larger than the current best integer solution. In fact, any feasible solution to the original MIP problem can be used as upper bound for the objective value of the tree nodes. In each node, a branching variable has to be selected.

Both Abela et al. [4] and Ernst et al. [34] select the branching variable based on the largest violation of the separation constraint in the current solution.

Ernst et al. introduce a problem specific version of the simplex algorithm to solve the LP problems in each node of the tree. They also apply extensive pre-processing on the formulation of the root problem and node problems. This seems to speed up the computation times considerably, compared to using a standard branch and bound algorithm.

Beasley et al. [17] restart the branch and bound algorithm multiple times, and tighten the landing time intervals in between. They use an initial upper bound on the problem by considering the flights in FCFS sequence. Using an upper bound on the objective, the landing time intervals for the flights can (possibly) be tightened in the following way: Let z_{ub} be the upper bound on the objective function. The landing time interval for flight i can be tightened in the following manner:

$$\begin{aligned} E_i &:= \max\{E_i, \hat{t}_i - \frac{z_{ub}}{A_i}\} \\ L_i &:= \max\{L_i, \hat{t}_i + \frac{z_{ub}}{B_i}\} \end{aligned}$$

If an improved integer feasible solution is found in the branch and bound algorithm within a short time, the objective is used as upper bound to tighten the time intervals. Then, the branch and bound algorithm is restarted.

Artiouchine et al. [9] use a MIP to solve the static problem with holding patterns and a fixed separation time S between every pair of successive flights. It is assumed that a flight i can land in an initial time interval $[E_{i1}, L_{i1}]$ and otherwise enters a holding pattern. After completing the holding pattern the flight can land in the interval $[E_{i2}, L_{i2}]$ or otherwise enters a holding pattern again. K_i landing time intervals are given for each flight i. The problem considered is a feasibility problem, since the objective is to maximize the minimum separation S obtained between all pair of aircraft, for which still a feasible schedule exists. If the optimal value for S is larger than the required separation between every successive pair of flights, a practically feasible schedule is found. The MIP formulation that is used to solve the problem considers a simple problem. The formulation does not have an objective function, but it is formulated to give a (possibly) preemptive schedule for a given value of S. It is found that this MIP usually can be solved within short time for practically sized instances. When the problem is solved and a (possible preemptive) schedule is obtained, an efficient algorithm is used to check if the solution also represents a feasible non-preemptive schedule. If this is the case, a solution to the original problem is found. If not, an additional constraint is formulated using the infeasible assignments and added to the MIP. This constraints avoids that the same (preemptive) schedule is obtained from the MIP again. This process is repeated until a non preemptive schedule is found. The optimal value for S is found by performing a binary search using the algorithm multiple times. Experiments were performed using a large number of randomly generated instances. All most all instances with 45 flights or less are solved within 100 seconds. However, the practical relevance of this model is limited, since the detailed consideration of holding patterns will be only useful on a short time horizon.

Heuristics

The solution techniques discussed until this point are (mainly) aimed at finding optimal solutions. These methods are only computationally tractable for small instances or under simplified assumptions. Heuristics are needed to solve larger instances. We will discuss the heuristics for the aircraft landing problem used in the literature in the remainder of this section. Considerable attention is paid to solving the problem with genetic algorithms in the literature. This class of heuristics is discussed separately.

Beasley et.al.[17] introduce a heuristic that (for the single runway case) is to just to land the flights in FCFS sequence. The objective is the weighted deviation from target landing times. The optimal landing times for the flights in this order are obtained by solving the LP formulation presented earlier. The heuristic gives mixed results. It is optimal for 4 of the 8 instances presented. But for three instances it gives large optimality gaps up to 75%. FCFS is the current procedure for landing flights, so this heuristic will not improve current practice.

Bianco et al. [21] introduce two heuristics, called the Cheapest Addition Heuristic (CAH) and Cheapest Insertion Heuristic (CIH). Earliest landing times (but no latest landing times) are considered and specific successive separation times between every pair of aircraft are considered. The objective is to maximize throughput.

In the CAH heuristic a flight is added to the end of the sequence in every iteration. Let H be the set of unscheduled flight and flight i be the latest scheduled flight at time t. Now consider the set of flights $J \subset H$ that contains the flights that have an earliest landing time, such that they can be scheduled after flight i without waiting longer than a time period that could be used to land another unscheduled flight:

$$J := \left\{ j \in H : E_j - (t + S_{ij}) < \min_{k,l \in H \setminus \{j\}} \{S_{kl}\} \right\}$$

If this set is nonempty, flight $j \in J$ for which S_{ij} is scheduled next at time $\max\{t + S_{ij}, E_j\}$. Otherwise the flight $k \in H$ for which E_k is minimum is scheduled next. This heuristic can be performed in $O(N^2 \log N)$ time.

In the CIH heuristic an unscheduled flight is simply inserted at the best position in the current sequence. In every iteration the best position for each unscheduled flight in the current sequence has to be determined. The total heuristic runs in $O(N^4)$ time. The quality of the solutions found by using

CIH is generally better than by using CAH. The algorithm is tested with two instances of the aircraft landing problem with 30 and 44 flights, respectively. The total landing time is decreased with 9% and 20%, respectively, by the heuristic compared to the FCFS solution. The gap with a derived lower bound on the optimal value is 4% and 13%, repectively for the two instances.

Erzberger [35] presents an algorithm for the real-time scheduling of landings. The application is to assist approach controllers in determining a feasible sequence for landing the flights. The algorithm assumes the aircraft enter the approach area at given times. This gives constraints on the landing order of the flights, because separation between flights has to be maintained. Additionally the distance from the approach area entry point and the speed of the aircraft can differ and therefore the flight times to the runway differ. The algorithm calculates the earliest possible landing times, considering these times and constraints. These times will determine the landing order, which can thus be slightly different from the FCFS order. The landing times are determined by landing the flights in the determined order as early as possible, respecting the required separation. This algorithm is used in a simulation study by Carr et al. [26] to evaluate the effects of airline prioritization. The airlines are allowed to provide a preferred arrival order for their own flights. In the initial order (in which the flights enter the approach area), the positions of these flights are exchanged. In this way, every airline keeps the same set of positions. In the experiments the hub airline supplies preferences and the other airlines do not. The results show that the average delay for the flights of the hub airline and the other airlines increase by considering these preferences. This increase is over 10% when the preferences include flights early in the schedule.

Anagnostakis and Clarke [5] introduce a two-stage heuristic for the scheduling of departures on a runway. Because of the similarities to the aircraft landing problem, the heuristic is discussed here. In the first stage only the weight categories are considered and a sequence of weight categories, such that the throughput is maximized (considering weight category based separation times) is determined. This is done by randomly generating a large number of sequences and evaluating their throughput. First the sequence with the largest throughput is selected. In the second stage individual flights of the appropriate category are assigned to positions in the sequence, considering departure time intervals. Additional constraints such as a Maximum position shift and flight pair specific separations can also be considered. This problem is formulated as a MIP problem and solved by a branch and bound algorithm. If there is no feasible solution for the current sequence, the second best sequence is considered. Simulation experiments based on data from Boston Logan airport were performed. In each simulation run 15 flights were considered. The results showed a 9% decrease in average departure delay and a 3% throughput increase, compared to a FCFS schedule. The computation time of this algorithm will be considerable for larger number of flights, because it involves solving an MIP problem (possibly multiple times).

It seems hard to strike a good balance between computation times and solution quality. The heuristics in [17] and [26] are overly simple and can be performed almost instantly but often fail to improve the FCFS schedule. The CIH heuristic in [21] performs relatively well, but is designed to use throughput as objective. Therefore, it cannot be expected to give equally good results with flight specific objectives (such as deviation from a target landing time).

Genetic Algorithms

Several papers use a Genetic Algorithm as heuristic to solve the ALP. Genetic Algorithms are inspired by biological evolution theory. Successive generations of individuals are evaluated. Fit individuals are more likely to survive. In genetic algorithms, individuals are encoded as chromosomes which represents solutions. The fitness of a solution can be measured by an objective function. New population members (children) are obtained by combining current members (solutions) using a crossover procedure. After crossover, mutation of the resulting solution can be applied. The general structure of a genetic algorithm is summarized below.

```
GENETIC ALGORITHM()
Generate initial population
for number of iterations
    do Calculate fitness value for each member of the current population
       Select parents from the current population
       Apply crossover operators to the selected parents to obtain children
       Apply mutation operators to improve and diversify the children
       Replace some members of the current generation with the children
return Best solution found
```

Usually, randomness plays a big role in these algorithms. It is often used in generating the initial population, selecting parents, the crossover and mutation operations and in replacing current members of the population. The representation of a solution by chromosomes, and the exact procedure used in each step of the algorithm differ between the papers discussed.

Abela et al. [4] introduced a genetic algorithm for solving the static aircraft landing problem with landing time intervals. The chromosomes represent an array of landing times for the flights. The objective is to minimize the deviation from the target landing time.

The initial generation consists of randomly generated solutions. For each child in a new generation, two parents from the previous generation are selected. The probability of selection is larger for parents with a better fitness value. The children are created by using the landing times of the landing time array of the first parent up to a random position and the landing times of the second parent from that position on. This could lead to an infeasible solution with respect to the separation constraints. A heuristic is used to update the times such that they are feasible, while keeping the order of the flights the same. Another heuristic is used to lower the cost of the schedule, while keeping the order of the flights the same. Random mutation is applied with a certain probability. This is done by randomly changing one of the landing times in the solution.

The algorithm was tested using 15 instances with up to 20 aircraft. 1000 generations of the genetic algorithm were completed for each instance. The average gap from the optimal solutions was 6%, with a worst case of 14%.

Beasley, Sonander and Havelock [19] also use a genetic algorithm for solving the static aircraft landing problem with landing time intervals. Their chromosomes represent an array of N real numbers y_i. This number represents the proportion of the landing time interval $[E_i, L_i]$ for flight i that elapse before aircraft i lands: $t_i = E_i + y_i(L_i - E_i)$. The crossover procedure is to randomly choose y_i from one of the parents for each position. No mutation was applied. The fitness (objective) function is peculiar: For each aircraft, the fitness f_i is calculated as follows:

$$f_i = \begin{cases} -(t_i - \hat{t}_i)^2 & \text{if } t_i \geq \hat{t}_i \\ (t_i - \hat{t}_i)^2 & \text{otherwise} \end{cases}$$

This reflects a preference to land an aircraft as early as possible and to penalize large delays from the target times disproportionally. The fitness of a solution is obtained by summing over the individual flight fitness values.

A unfitness value is defined, to penalize violations of the separation constraints for every pair of aircraft i and j, by $\max\{0, S_{ij} - (t_j - t_i)\}$. The total unfitness value is obtained by summing over the all the individual unfitness values.

The fitness and unfitness values are used to replace a member of the current population with a child. If there are members in the current population that have a worse fitness as well as a worse unfitness than the child, one of those members is randomly replaced by the child. Otherwise, if there are members in the current population that have a worse unfitness than the child (but a better fitness), one of those members is randomly replaced by the child. If this is not the case, it is checked if there are members that have worse fitness (but a better unfitness) and one of those members is randomly replaced by the child. If none of this was true, one of the members with better fitness and better unfitness is replaced by the child.

The algorithm is tested using a dataset of 20 aircraft describing the situation at a particular day and time at London Heathrow. For this dataset it is able to find solutions that increased throughput up to 5% and decreased average delay (wich was already low), compared to the FCFS sequence.

Ernst et al. [34] combine an addition heuristic and a genetic algorithm to solve the static problem with landing time intervals. The objective is the weighted deviation from target times. The heuristic is used to determine a landing sequence for the flights. In each iteration a flight is added at the end of the sequence. In each iteration k, for every unscheduled flight i a priority is calculated as

$$a\hat{t}_i + b\hat{E}_i^k + \alpha_i$$

with $\hat{E}_i^k = \max\left\{E_i, \max_j\{\hat{E} + S_{ji}\}\right\}$ with j the flights scheduled in earlier iterations. In the experiments $a = 8$ and $b = 1$ is used. α_i is a value used to perturb the solution. The flight with the lowest priority value is added to the sequence. The optimal landing times of the flights given this sequence are obtained by solving the LP formulation from the previous section.

The chromosome of the genetic algorithm is an array of the perturbation values α_i. These are randomly drawn from an exponential distribution. To obtain a child, single-point crossover is used. The child replaces the worst member of the current population. The heuristic was used with 10 instances with up to 50 flights. For every instance the heuristic was performed 20 times. For 8 instances the optimal solution was found every time. The average gap over the 20 times was 1% and 3% for the other two instances. The worst gap that accured was 11%.

Genetic Algorithms seem capable of finding good solutions fast, even for large instances. A drawback is the randomness involved in these algorithms. This causes variance in the quality of the solutions. The uncertainty with respect to the quality of the solutions can be a problem for practical use of these types of algorithms.

3.1.4 Multiple Runway Problems

In multiple runway problems, next to determining a landing sequence and time, each flight has to be assigned to one of the runways. Depending on the runway system layout, separation constraints can also apply between aircraft landing on different runways.

The multiple runway problem has also received considerable attention in the literature. Some of the single runway models were extended to the multiple runway case.

In Psaraftis [58] the two runway problem was considered. It uses the dynamic programming formulation from Psaraftis [59], which was given in the previous section. All possible partitions of the groups of aircraft between the runways are simply enumerated for the two runway case.

Beasley et al. [17] extended their single runway MIP formulation that was presented in the previous section. Let R be the number of runways that can be used. Additional decision variables are introduced:

$$\begin{array}{lcll} \zeta_{ij} & = & \begin{cases} 1 & \text{if flight } i \text{ and } j \text{ use the same runway} \\ 0 & \text{otherwise} \end{cases} & i,j \in F, i \neq j \\ \gamma_{ir} & = & \begin{cases} 1 & \text{if flight } i \text{ uses runway } r \\ 0 & \text{otherwise} \end{cases} & i \in F, r = 1,\dots,R \end{array}$$

And the following constraints to obtain a feasible runway assignment are added:

$$\begin{array}{rcll} \sum_{r=1}^{R} \gamma_{ir} & = & 1 & i \in F \\ \zeta_{ij} & = & \zeta_{ji} & i,j \in F, j > i \\ \zeta_{ij} & \geq & \gamma_{ir} + \gamma_{jr} - 1 & i,j \in F, j > i, r = 1,\dots,R \end{array}$$

Separation constraints can also apply between aircraft landing on different runways. Let S^m_{ij} be the separation required if flight i and j land on different runways. The updated separation constraint becomes:

$$t_j \geq t_i + \zeta_{ij} S_{ij} + (1 - \zeta_{ij}) S^m_{ij} - M\delta_{ji} \qquad i, j \in F, i \neq j$$

where M is a sufficiently large positive constant.

Beasley et al. also give a simple heuristic to solve the multiple runway problem. The flights are ordered by their target landing times. The flights are each assigned to a runway in this order in the following way: Let t_{ir} be the maximum of the target time of flight i or the earliest time this flight can land on runway r given the separation constraints:

$$t_{ir} := \max\left\{\hat{t}_i, \max_{j<i}\{t_j + \gamma_{jr} S_{ji} + (1 - \gamma_{jr}) S^m_{ji}\}\right\}$$

Flight i will be assigned the runway u with $u := \arg\min_r\{t_{ir}\}$ with landing time $t_i = t_{iu}$ and $\gamma_{iu} = 1$. This heuristic is used to determine the runway assignment of the flights. The sequence is still determined by their target times. The optimal landing times are recalculated afterward by solving the LP with the runway assignment and sequence variables fixed. The heuristic gives mixed results, giving the optimal solution for 13 of the 17 multiple runway instances tested. However, most of these instances represent situations where capacity (largely) exceeds demand and almost every flight can land at its target time. In 4 instances where capacity is more constrained, optimality gaps between 40% and 90% occur. The heuristic solution is also used as an upper bound on the optimal solution in the branch and bround algorithm.

Beasley, Krishnamoorthy et al. [18] extended the genetic algorithm of Beasley, Sonander and Havelock [19] to include runway assignment. Separation constraints are only considered between flights landing on the same runway. The problem is considered in a dynamic setting. Each flight has an appearance time. The problem is first solved for the flight(s) with the earliest appearance time. The problem is resolved when a new flight appears. When the problem is resolved, the objective includes displacement cost for scheduling the flight further away from the target time than in the previous solution. On average, optimality gaps are small (around 11%) and the savings compared to FCFS are large.

Pinol and Beasley [56] use scatter search and a bionic algorithm as heuristics to solve the multiple runway problem. Both algorithms have similarities with genetic algorithms. Separation constraints between flights landing at different runways are considered. Next to this, runway dependent landing time intervals and separation times are considered. The basic representation is the same as in Beasley, Sonander and Havelock [19], which was discussed in the previous section. Next to the landing time interval variable y_i, the runway r_i that flight i is assigned to is represented in the solution chromosome. Both (linear) deviation from the target landing time and the peculiar objective function as used in [19] to land an aircraft as early as possible and penalize large delays from the target times disproportionally are considered.

Scatter search combines multiple parents to one child. In this case 3 parents are used. Each parent is selected, by choosing 2 solutions from the current generation at random and select the one with the best fitness. The child replaces the worst member of the current population.

Bionic algorithms aim to have a population which a diverse set of solutions. This diversity is measured by a distant measure between solutions. For each flight, the distance is 1 if the flight lands on a different runway in each solution and the absolute difference in y_i value between the two solutions otherwise. The total distance between the solutions is the sum over the flight differences. A graph is repeatedly build where solutions are represented as nodes. The fitness value of a solution determines how often it is included in a graph. Nodes are connected by an edge if their distance is smaller than a predefined threshold. Parents are selected by selecting a maximal independent set of nodes from the graph (meaning that solutions that are not connected are selected together). This process is repeated several times, to generate a set of new members. The best member of this set is selected and replaces the worst member of the current population.

In both algorithms, the value y_i of a child is a weighted linear combination of the parents' y_i values. The weights for each parent are drawn randomly such that the sum of the weights is 1. The runway r_i is chosen from one of the parents at random. The solution of the new child is improved by computing the optimal landing times given the runway assignments and sequences.

Experiments are performed using 13 instances with 10 to 500 flights. Each instance is tested with using up to 5 different runways. Because of the randomness, the heuristics are performed 10 times for each combination. Only the results of the best of those 10 runs are reported. When the nonlinear objective is used the gap for these solutions compared to the optimal solutions is on average below 2% for both algorithms. It is remarkable that for the (larger) single runway instances (which are more capacity restricted) the gaps are larger, with values up to 19%. The improvement compared to the FCFS schedule is usually around 5%. For the linear objectives the optimality gaps are larger (up to 52% for the single runway instances). The improvement using the bionic algorithm compared to the FCFS schedules for the congested single runway problems is also larger for this objective (around 15%). The scatter search has a better performance on the multiple runway problems. The computation times for the large instances (500 flights) were around 10 minutes.

Ernst et al. [34] also extend their problem specific branch and bound algorithm and genetic algorithm for the multiple runway case. The multiple runway case of the genetic algorithm adds the runway of each flight to the chromosome. For the initial population, these are randomly generated. Ten instances with up to 50 flights combined with 2,3 or 4 runways are tested. In total 22 of these combinations, from which 13 are solved optimally by the GA in every run. The optimal solution for each solution is found by the GA in at least one of its runs. However, runs with large gaps of over 100% also occur.

3.1.5 Discussion

The Aircraft Landing Problem (ALP) is the problem of determining a landing sequence for a set of aircraft and a landing time for each aircraft, respecting the required (sequence-dependent) separation between flights. Most models consider the objective of maximizing runway throughput or minimizing flight delay metrics. These metrics consider all flights as interchangeable. This does not reflect the reality of the airlines, for which the impact of a delay differs from flight to flight. Although some of the models discussed use flight dependent weights in the objective function, no mature approach that considers airline preferences and the related fairness issues was found in the ALP literature.

The ALP is a complex problem for which it is difficult to find optimal solutions for larger instances within reasonable computation time. A wide range of solution techniques have been considered for the aircraft landing problem. These solution techniques represent different trade-offs between solution quality and computation time. Approaches based on (complete) enumeration, including dynamic programming and mathematical programming, are aimed at finding an optimal solution. These methods are only computationally tractable for small instances of the problem. Heuristics are needed to solve larger instances. Some of the heuristics discussed are overly simple and can be performed almost instantly but often fail to improve the FCFS schedule. Other heuristics perform relatively well. Most heuristics are designed with a specific objective in mind and are not (directly) applicable when another objective is considered. Genetic algorithms seem capable of finding good solutions fast, even for large instances. These algorithms are flexible for use with different objectives. A drawback is the randomness involved in these algorithms. This causes variance in the quality of the solutions. The uncertainty with respect to the quality of the solutions can be a problem for practical use of these types of algorithms.

3.2 Ground Delay Programs

Ground delay programs are related to the aircraft landing problem, in the sense that they assign airport runway capacity to landing flights. The approach is different, because runway capacity is not optimized but provided as input. The goal of ground delay programs is to distribute landing slots in a cost-effective and fair way to the flights and airlines.

As explained in Section 2.3.2, a Ground Delay Program (GDP) is issued when it is expected that (later on that day) airport arrival demand will exceed capacity for a sustained period of time (e.g., because of severe weather conditions) at an airport. To balance demand and capacity delays have to be assigned to flights planning to land at the airport in the considered period. During a ground delay program these delays are (as much as possible) assigned to flights on the ground at their origin prior to departure rather than en-route, which is both safer and cheaper.

The (expected) airport arrival capacity during the period of congestion has to be estimated when a GDP is issued. It is usually not known in advance how long the reduced capacity period will last, because this depends on weather conditions that are hard to predict. This complicates the problem: If the capacity is overestimated, expensive air delays will still occur. When capacity is underestimated, passengers are unnecessarily delayed and scarce airport capacity goes unused. To hedge against the uncertainty in the airport capacity, long haul flights are usually exempted from a GDP. This limits the capacity that goes unused when the capacity reduction ends earlier than expected. If the capacity reduction lasts longer than expected short haul flights can still be assigned ground delays to avoid expensive air delays.

A GDP can also be dynamically revised during execution if the weather (forecast) changes.

A number of landing slots equal to the estimated capacity are defined. These landing slots are allocated to flights that were originally scheduled during the period of congestion. By subtracting the flight times from the assigned landing times, updated departure times are determined for the flights. Thus, ground delays at the origin are assigned to these flights. It has to be determined how to allocate the slots and with that how much (ground) delay to assign to each flight.

In this section, mathematical models considering the assignment of flights to slots in a GDP will be discussed. The representation of (estimated) airport arrival capacity is an important aspect of these models. In current practice, capacity is treated as a deterministic value. To represent the uncertainty, capacity can also be represented by a number of scenarios each with a certain probability of occurring. The problem can be considered in a static or dynamic manner. The latter explicitly considers the fact that a GDP can be revised during execution when updated capacity forecasts become available. Several papers consider GDP procedures under collaborative decision making, as currently used in practice in the U.S. The interactions occurring when ground delay programs are active at multiple airports at the same time are also studied in the literature, for example by Andreatta et al. [7]. This subject is however beyond the scope of this thesis.

In the remainder of this section the different types of models are discussed in detail, followed by a more general discussion of the GDP literature.

Static Deterministic Models

The static deterministic problem can be formulated as a network flow problem, as was done by Terrab and Odoni [75]. Network flow problems are well studied and efficient solution methods exists.

The considered time period is divided into T smaller time periods (e.g., 15 minutes). The airport capacity for each time period t is given by C_t. Let $F = \{1, \ldots, N\}$ be the set of flights to schedule. Let x_{it} be 1 if flight i is assigned to time interval t and 0 otherwise. Let $\hat{t}_i$ be the time interval in which flight i was originally scheduled or expected to land. Let c^g_{it} be the ground delay cost of assigning flight to time period $t \geq \hat{t}_i$.

The objective to minimize total ground delay cost becomes:

$$\min \sum_{i \in F} \sum_{t=\hat{t}_i}^{T} c_{it}^{g} x_{it}$$

The following constraints enforce that each flight is assigned to a single time period and the capacity is not exceeded:

$$\begin{aligned} \sum_{t=\hat{t}_i}^{T} x_{it} &= 1 \qquad i \in F \\ \sum_{i \in F} x_{it} &\leq C_t \qquad t = 1, \ldots, T \end{aligned}$$

Terrab and Odoni use cost functions $f_i(k)$ to describe the cost of assigning k time periods of ground delay to flight i:

$$c_{it}^{g} := f_i(t - \hat{t}_i)$$

They give a very efficient algorithm to obtain optimal solutions for this model under some regularity conditions for the cost functions $f_i(k)$.

Hoffman and Ball [42] extend the model by including banking constraints. These constraints represent the desire of airlines to land banks of flights within fixed time windows to accommodate the hub operations of airlines. These constraints make the problem computationally harder to solve. Different MIP formulations of the problem with banking constraints are evaluated both computationally and analytically.

Static Stochastic Models

The static stochastic model from Richetta and Odoni [62] was one of the first models proposed for the GDP. In this integer programming model, Q different capacity scenario's are considered. Scenario q has probability p_q and C_t^q is the landing capacity in period t under scenario q. The models are static in the sense that all ground delays are assigned at the time the GDP is issued and will not be revised. At this time it is not known which capacity scenario will occur in reality. Therefore, it might be necessary to assign air delays under some scenarios in order to balance demand and capacity.

The model does not consider individual flights but aggregates flights by the time interval they were originally scheduled to land. Let N_s be the number of flights that was originally scheduled in time interval s. The decision variables x_{st} represent the number of flights that were originally scheduled to arrive in time interval s but are assigned a ground delay of $(t-s)$ time intervals and thus will land in interval $t \geq s$. Decision variables w_t^q represent the number of flights that are delayed in the air from period t to period $t+1$ under capacity scenario q.

The function $f(k)$ represents the cost of delaying a flight for k periods on the ground. The marginal cost of delaying a flight in the air for one period is assumed to be constant. This constant is denoted by c^a. A common and realistic assumption is that the marginal cost of air delay are larger than the marginal cost of ground delay:

$$c^a > f(k+1) - f(k) \qquad k = 1, \dots, T-1.$$

The objective is to minimize the sum of the ground holding cost and the expected air holding cost:

$$\min \sum_{s=1}^{T} \sum_{t=s}^{T} f(t-s) x_{st} + \sum_{q=1}^{Q} p_q \sum_{t=1}^{T} c^a w_t^q$$

The following constraints enforce that every aircraft is landed and the capacity is not exceeded under any of the scenarios:

$$\begin{aligned}
\sum_{t=s}^{T} x_{st} &= N_s && s = 1, \dots, T \\
\sum_{s=1}^{T} x_{st} + w_{t-1}^q - w_t^q &\leq C_t^q && q = 1, \dots, Q, t = 1, \dots, T \\
x_{st}, w_i^q &\geq 0 \text{ and integer}
\end{aligned}$$

Kotnyek and Richetta [48] show that if $f(k)$ is monotonically increasing in k, solving the LP relaxation guarantees an integer solution and thus in that case the problem can be efficiently solved.

Terrab and Odoni [75] propose a similar stochastic model considering individual flights, similar to their static model mentioned earlier. They propose a dynamic program and several heuristics to solve the model. They compare the deterministic and stochastic models and identify that substantial decrease in delay cost can be obtained by considering the stochastic nature of capacity.

Dynamic Stochastic Models

Richetta and Odoni [63] extend their static stochastic model to a dynamic stochastic model. The capacity scenarios are represented as a conditional probability tree. This tree divides the time horizon in a number of stages. A stage begins at a time when the tree branches. When time progresses the conditional probability of future capacity scenarios change. At the beginning of each stage the conditional probability of future capacity given the current capacity is updated and ground delays are assigned to flights departing during this stage. The optimal ground delays are obtained by solving a multi-stage stochastic integer program.

Richetta [61] introduces a heuristic to solve the problem in a more efficient manner. At the beginning of each stage the static deterministic problem is solved, considering the most likely capacity scenario (according to the current conditional probabilities) and fixing ground delays for flights departing during earlier stages.

Computational experiments are performed based on data from Boston's Logan International airport and compared to static and dynamic optimal solutions. The results show a 25% reduction in delay cost on average when comparing the optimal dynamic stochastic solution to the optimal static stochastic solution. The amount of air delay necessary decreases substantially when the dynamic model is used. The heuristic obtains very good results, achieving almost all of the potential delay reduction.

Mukherjee and Hansen [55] extend this model by allowing revisions to previously assigned ground delays to flights that have not yet departed. Resulting in an additional 11% delay cost reduction on average.

Collaborative Decision Making

As Vossen and Ball [76] state, the models discussed above can be said to follow a central planning paradigm, in that system-wide optimal solutions are developed without considering the impact on individual airlines. It is difficult to apply these models under the CDM paradigm.

Vossen and Ball [76] discuss the GDP procedures under CDM that are currently used in practice in the U.S. (see Section 2.4.1) in a formal manner by mathematical models. The process consists of the following steps:

1. The number of arrival slots to assign is determined based on the estimated capacity.

2. Arrival slots are allocated to *airlines* (instead of individual flights).

3. Each airline decides on an allocation of its flights to the slots it received.

4. Unassigned (released) slots are reallocated.

Below (alternative) models for these steps as proposed in the literature are discussed.

Currently the expected capacity is used to determine the number of available slots. This means that the uncertainty in the capacity forecast is not considered. Ball et al. [13] propose a model that calculates the optimal number of slots s_t to allocate in time interval t considering probabilistic capacity scenarios, the number of flights originally scheduled in each time interval and the cost ratio λ between marginal air and ground delay cost. Let x_t be the number of flights that are delayed on the ground from period t to $t+1$. The objective of the model is:

$$\min \sum_{t=1}^{T} x_t + \sum_{q=1}^{Q} p_q \sum_{t=1}^{T} \lambda w_t^q$$

The following constraints enforce that every aircraft is landed and the capacity is not exceeded under any of the scenarios:

$$\begin{array}{rcll} \sum_{t=s}^{T} s_t - x_{t-1} + x_t & = & N_s & s = 1, \ldots, T \\ \sum_{s=1}^{T} s_t + w_{t-1}^q - w_t^q & \leq & C_t^q & q = 1, \ldots, Q, t = 1, \ldots, T \\ s_t, x_t, w_i^q & \geq & 0 \text{ and integer} & \end{array}$$

The model is a simplification of the static stochastic model of Richetta and Odoni [62]. This simplification is motivated by CDM procedures, in which the slots are not assigned to individual flights and thus it is sufficient to determine the optimal number of slots to assign (per time period). This can also be done by the Richetta and Odoni model, with $\sum_s x_{st}$ as the optimal number of slots for time period t. Kotnyek and Richetta [48] compare both models.

Under CDM, arrival slots are allocated to airlines instead of individual flights. The initial assignment of slots is done by a procedure called *ration* *by schedule*. This procedure orders all flights according to their original schedule time and assigns them to the slots in this order. This assignment can be obtained by solving the deterministic model of Terrab and Odoni [75] with $f_i(t - \hat{t}_i) := (t - \hat{t}_i)^\sigma$ for all flights i, $\sigma > 1$ and $\hat{t}_i$ the original schedule time of flight i.

Abdelghany et al. [3] consider the problem of an airline to allocate its flights to the slots it received. The goal considered is to minimize the overall downline impact resulting from the allocation. A genetic algorithm is used to solve the problem.

Airlines can also decide to cancel flights and thus release slots. This might be necessary if flights are already delayed beyond their original schedule time and will not be able to use the assigned slots. A procedure called *compression* is used to fill these unassigned slots. The idea behind compression is to reward airlines for slots they release, thus encouraging airlines to report delays and cancellations. This is done by allocating a replacement slot (as early as possible) in return for the unused slot. This reassignment is currently performed by the FAA, using a fixed procedure. Alternative procedures for the compression method involving a shift to decision-making by the airlines are proposed in the literature. Vossen and Ball ([76] and [77]) propose models in which airlines can propose requests for slot exchanges which are considered by a mediator (the FAA). According to their results, this improves on-time performance and passenger delay measures. The mediating process is designed with fairness considerations in mind.

In his PhD thesis, Hall [40] proposes an alternative to the current CDM procedures by proposing an auction of GDP arrival slots. Airlines supply a value for each flight/slot combination. An assignment problem is solved with the objective to maximize the total value. In order to get truthful values, each airline has to pay a fee that is equal to the value lost by the other airlines caused by the presence of the considered airline. The author concludes that although in theory the potential benefit is enormous, it would be complicated to implement such a method.

Hall [40] also proposes an extension of the current CDM procedures by considering airport arrival and departure capacity (and their interdependence) at the same time. Airlines receive *runway operation* slots using a ration by schedule procedure. Each airline can decide to use an assigned slots for either an arriving or a departing flight. An extensive simulation study shows that airline oriented performance measures improve compared to traditional ground delay programs, especially on low capacity scenarios.

3.2.1 Discussion

In a GDP landing slots and with that ground and air delays are assigned to flights. The models discussed in this section show that considering the uncertainty in capacity estimates and dynamically revising of assigned ground delays can decrease the total delay cost substantially compared to static deterministic approaches.

The current application and models of GDP under CDM consider airline preferences by allowing individual airlines more freedom in allocating their flights to arrival slots. Because of fairness (and feasibility) considerations these assignments are constrained by the initial assignment using the ration by schedule procedure and the willingness of airlines to engage in (inter-airline) slot substitutions. These constraints can degrade the overall quality of the resulting allocation.

3.3 Contributions

In this chapter, literature considering the aircraft landing problem and ground delay programs were discussed. Both problems consider the problem of allocating airport runway capacity to flights.

In the aircraft landing problem the sequence dependent separation required between landing aircraft is used to obtain efficient landing sequences. Most models discussed measure efficiency in terms of runway throughput or flight delay metrics. These metrics consider all flights as interchangeable. This does not reflect the reality of the airlines, for which the impact of a delay differs from flight to flight. Although some of the models discussed use flight dependent weights in the objective function, no mature approach that considers airline preferences and the related fairness issues was found in the ALP literature.

In the GDP literature there is considerable attention for considering airline preferences. In a GDP landing slots are allocated to flights. The preferences of airlines are considered by allocating slots to airlines (instead of individual flights). This initial assignment procedure is performed centrally and is designed with fairness considerations in mind. After this initial assignment, slot substitutions can be performed by the airlines. However, these slot substitutions procedures are constrained by the initial assignment and the willingness of airlines to engage in (inter-airline) slot substitutions. These constraints on slot substitutions are in place in order to achieve a fair allocation, but can degrade the overall quality of the allocation.

The slots considered in ground delay programs have equal length. The sequence dependent separation required between landing aircraft and the possibilities this gives to increase runway throughput are not considered. This can result in inefficient use of scarce runway capacity.

The goal of our research is to fill the gap between these two approaches.

The approach of this research is similar to the aircraft landing problem in the sense that runway operations times are scheduled considering the sequence dependent separation. In this way, possible increases in runway throughput obtained by sequencing the flights are considered. This is a clear advantage compared to ground delay programs.

The aggregate efficiency metrics (such as throughput), which are usually considered in ALP models, are not used as primary objective in this research. An objective based on airline cost metrics is used to evaluate the quality of a schedule. This is done in order to reduce the impact of delay to airlines and their passengers. A novel approach to represent airline cost and incorporate those in a fair manner in the scheduling process is developed. This approach forms a contribution relative to existing ALP models.

In ground delay programs airlines preferences are incorporated by allowing airlines to decide on slot substitutions. This means some decisions are decentralized. Our approach is different in the sense that the decisions are centralized: Airlines are requested to communicate their preferences to a central decision maker, who uses these preferences in establishing a runway operations schedule. There are two reasons for this approach:

In ground delay programs severe constraints on slot substitutions are in place in order to achieve a fair allocation. These constraints can degrade the overall quality of the resulting allocation. These constraints are not needed in our approach. This does not mean fairness is not considered in the research, but a central decision maker can weigh fairness and overall schedule quality in a less constrained manner. This forms another advantage compared to ground delay programs.

The second reason is that slot substitutions become complicated when considering sequence dependent separation. The runway operation time associated with slots could change when altering the sequence of the flights. This complicates the possibility to allow airlines to decide on slot substitutions, because this could affect flights not involved in the slot substitutions.

Chapter 4

Airline Cost and Fairness

The goal of this research is to develop an approach to consider airline preferences in runway operations scheduling to reduce the impact of delays for airlines and their passengers. A difficulty when considering individual airlines preferences is fairness. It is the role of air traffic control to assure that air traffic proceeds in a safe, efficient and equitable manner. Consequently, scarce air traffic capacity has to be assigned to competing airlines in a fair manner. The approach presented in this chapter, consists of a representation of airline preferences and a procedure to obtain a fair and efficient schedule using these preferences. This approach was originally introduced in Soomer and Franx [71] and will be discussed in Section 4.1.

The airline preferences are represented by a cost function for each flight. This cost function is used to relate the runway operation time of the flight to the cost incurred by the airline. We want to allow the airlines as much flexibility as possible in representing these cost functions. At the same time, these cost functions must be applicable to establish a fair and efficient runway schedule. Therefore, it must be possible to compare the cost functions from competing airlines in a fair manner. Additionally, it should not be possible for airlines to conduct strategic behavior. To achieve this, a combination of centralized decision making, restrictions on the shape of the cost functions and a cost function scaling mechanism are proposed.

In Chapter 5 and 6 optimization models and algorithms are presented that can be used by the decision maker to obtain a fair and efficient schedule. A straightforward objective to consider in determining a schedule is to minimize the total scaled cost. This can be considered fair in the sense that all flights are treated in the same manner and the cost of different flights can be fairly compared independent of their airline (using the scaled cost functions). However, it does not necessarily lead to a schedule in which the scaled cost and/or delay are shared equally (or proportionally) among the airlines. In order to evaluate the fairness of an obtained schedule, three definitions of fairness are presented in Section 4.2. Related objectives that can be used during the optimization process and have an explicit focus on fairness are also presented. These definitions were originally presented in Soomer and Koole [72].

4.1 Airline Cost Representation

In runway operations scheduling, a schedule with a feasible runway operation time (landing or take-off) for each flight is established. Airlines have a preferred runway operation time for each of their flights. A deviation from this preferred time, will incur (additional) costs for the airline. This explains our approach to represent airline preferences by cost functions. A cost function is used to relate the runway operation time of a flight to the cost incurred by the airline. In the remainder of this chapter, the term runway operation time will be replaced by the term landing time for the readers convenience.

In order to consider the airline preferences as directly as possible, we want to allow the airline as much flexibility as possible in representing these cost functions. At the same time, it must be possible to obtain a fair and efficient schedule using these cost functions. In order to achieve this, it must be possible to compare cost (functions) from different airlines in a fair manner. Additionally, it should not be possible for airlines to conduct strategic behavior. In other words, the cost functions should only depend on the airlines own preferences. It is not wanted that airlines can gain a structural advantage over other airlines by intentionally misrepresenting their preferences. A common approach to avoid strategic behavior is to apply a market mechanism (usually involving payments). However, this is complicated in this case. Runway operations scheduling is related to tactical and operational air traffic activities. This means the time horizon is limited and priority at this point in time is on safety and efficiency. This makes it hard to guarantee an exact landing time (in return for the payment) in advance. Furthermore, airlines will object to (additional) payments required for the use of airspace and airport capacity.

Therefore, another approach is chosen to ensure that cost (functions) from different airlines can be compared in a fair manner and to avoid strategic behavior from the airlines. This approach consists of a combination of centralized decision making, restrictions on the shape of the cost functions and a cost scaling mechanism.

Centralized Decision Making

In order to engage in strategic behavior successfully, an airline needs a good estimation of the cost functions of the other airlines. In the centralized decision making approach proposed here, airlines are requested to communicate their cost functions to an air traffic manager. These cost functions will not be shared with the other airlines and thus the possibilities to conduct strategic behavior are reduced.

Cost Functions Shape Restrictions

It is more convenient to compare cost functions from different airlines in a fair manner if they have a similar shape. The restriction on the shape of the cost functions will also limit the possibilities to conduct strategic behavior. The functions are required to be convex and piecewise linear and to have a minimum cost of zero at the preferred landing time. Thus, the non-negative cost functions are in fact describing the (additional) cost incurred by the airline for deviations from the preferred landing time. An example of such a cost function is depicted in Figure 4.1.

Convex functions are a realistic description of airlines costs related to delays. The convex shape represents an increase in the cost *per minute delay* (marginal cost of delay) with the total length of delay. This shape exactly meets the description of airline delay cost found in literature and by interviews with airline experts: A study performed by the Transport Studies Group of the University of Westminster and published by EUROCONTROL [28] estimated airline delay costs. The results show that the cost per minute delay increase with the total time of delay (and the number of passengers) of the flight. The average cost per minute of a short delay (around 15 minutes) range from below 1 euro to almost 50 euro depending on the number of (occupied) seats. Beatty et al. [20] introduce the concept of a delay multiplier to estimate the total system impact of a delayed flight. A delay multiplier is applied to the initial delay of an aircraft to estimate the amount of propagated delays to all flights connected to the initial flight by crew or by aircraft. The delay multiplier grows nonlinearly with the length of the initial delay. From interviews with experts from a major European Airline,

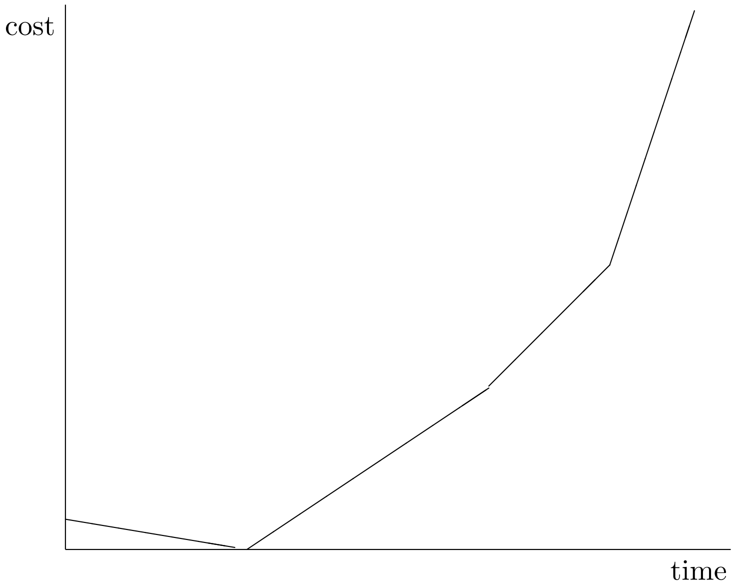


Figure 4.1: Example of a convex piecewise linear cost function

it became clear that their cost caused by arrival delays for flights at the hub airport are for a large part related to transfer passengers missing their connections. Of course the number of missed transfers will increase with the length of delay (until the point where all connections are missed). This shows that convex functions are a natural choice to represent flight delay cost.

The restriction on the function to be piecewise linear is a minor technical restriction. Any convex function can be accurately approximated by a (convex) piecewise linear function. However, convex *piecewise linear* functions are convenient to use in mathematical programming models, as will be considered in Chapter 5 and 6.

Let us describe a convex piecewise linear function $f(x)$ representing the cost of landing a flight at time x, more formally. The continuum of landing times can be be subdivided in $(K+1)$ time intervals:
$[0, X_1], [X_1, X_2], \ldots, [X_K, \infty)$, such that the costs in each of these intervals

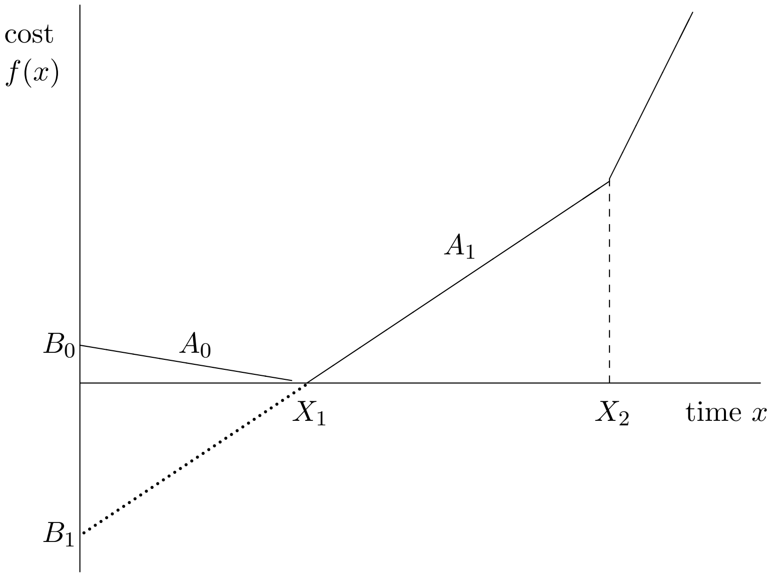


Figure 4.2: A convex piecewise linear cost function, illustrating the notation used

is represented by a linear function:

$$f(x) := \begin{cases} A_0x + B_0 & 0 \leq x \leq X_1 \\ A_1x + B_1 & X_1 \leq x \leq X_2 \\ \vdots & \vdots \\ A_Kx + B_K & X_K \leq x \end{cases}$$

A_k and B_k represent the slope and intercept, respectively, of the linear functions describing the cost between time X_k and X_{k+1}. $X_1, \ldots, X_K$ are denominated the K breakpoints of the function $f(x)$. This notation is illustrated in Figure 4.2.

In order to be convex, $f(x)$ has to be a continuous function. This is the case if the function is continuous in every breakpoint X_k, meaning:

$$A_{k-1}X_k + B_{k-1} = A_kX_k + B_k \qquad k = 1, \ldots, K. \tag{4.1}$$

Furthermore, $f(x)$ is convex if:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) \qquad 0 \leq \lambda \leq 1. \tag{4.2}$$

.

The cost functions must be non negative ($f(x) \geq 0$, $x \geq 0$) and and have a minimum of zero: $\exists x \geq 0 : f(x) = 0$.

These restrictions on the cost functions will still leave airlines with a lot of flexibility in representing the costs for a flight. This involves determining the preferred landing time (at which the costs are 0), the number and locations of the breakpoints and the slopes of the lines. The breakpoints can be seen as times at which cost will make a large step, or after which cost will increase faster than before. The slopes represent the marginal cost in the time intervals between two breakpoints.

Cost Scaling Mechanism

The submitted costs can vary a lot between different airlines. Now, minimizing the sum of all cost functions, will favor airlines that define relatively large cost for their flight delays. At first sight this seems reasonable, since this will lead to minimal total cost. However, we have to bear in mind that all airlines are allowed to define their own cost functions. Therefore, they will be able to obtain higher priorities for their flights, by (falsely) representing very large cost for delays. This leaves room for strategic behavior. This troublesome aspect can be overcome by rescaling the cost functions as provided by the airlines. This is done in such a way that each airline has a total amount of scaled cost at its disposal that is proportional to its number of flights and the possible landing times for these flights. In the scaling process the cost-ratio between flights of a single airline should be preserved, to reflect the economic trade-off for this airline. Therefore, the same scaling factor will apply to all flights of the same airline. The individual cost functions of the flights are multiplied by the scaling factor of their airline. Using the scaled cost functions, the cost of flights of different airlines can be compared in a fair way.

Let us make this more precise. First the following notation is introduced:

Let $F = \{1, \ldots, N\}$ be the set of all flights to consider.
Let A be the set of all airlines.
Let $F_a \subset F$ be the set of flights of airline $a \in A$. Note that $F = \bigcup_{a \in A} F_a$ and $F_a \cap F_b = \emptyset$, for all $a, b \in A, a \neq b$.
Let $\kappa_i(t)$ the cost function, relating the landing time with cost, for flight i.
Let $f_i(t)$ be the scaled cost function, relating the landing time with scaled

cost, for flight i.
Let E_i and L_i be the earliest and latest possible landing times of aircraft i, respectively.

Consider airline $a \in A$ with $|F_a|$ arriving flights, with convex piecewise linear cost functions $\kappa_i(t)$ for $i \in F_a$.

To obtain equity, these cost functions will be scaled to new cost functions $f_i(t) = \alpha_a \kappa_i(t)$ $(i \in |F_a|)$. The scaling factors α_a are determined per airline. This ensures that the ratio between costs of their own flights are preserved in the scaled objective functions:

$$\frac{f_i(t)}{f_j(t)} = \frac{\alpha_a \kappa_i(t)}{\alpha_a \kappa_j(t)} = \frac{\kappa_i(t)}{\kappa_j(t)} \qquad i, j \in F_a.$$

The scaling factor α_a is defined such that:

$$\frac{1}{|F_a|} \sum_{i \in F_a} \frac{\int_{E_i}^{L_i} \alpha_a \kappa_i(t)dt}{(L_i - E_i)^p} = 1.$$

So,

$$\alpha_a = |F_a| \left(\sum_{i \in F_a} \frac{\int_{E_i}^{L_i} \kappa_i(t)dt}{(L_i - E_i)^p} \right)^{-1},$$

where p is a parameter to minimize the effect of differences in the length of the landing intervals. For $p = 1$ the average scaled cost per time unit per flight for an airline will be equal to 1. It is preferable to choose p equal to 2 or just over 2, to give a small flexibility reward for airlines with flights with a relatively large average time interval.

Let us explain this. Consider two airlines with only one flight and identical cost functions $\kappa_1(t) = \kappa_2(t) = t$. The possible landing interval of flight 1 and 2 are $[0, T_1]$ and $[0, T_2]$ respectively, with $T_2 > T_1$. Now

$$\begin{aligned} \alpha_1 &= 2T_1^{(p-2)} \\ \alpha_2 &= 2T_2^{(p-2)} \end{aligned}$$

and the scaled objective functions are:

$$\begin{aligned} f_1(t) &= 2T_1^{(p-2)} t \\ f_2(t) &= 2T_2^{(p-2)} t. \end{aligned}$$

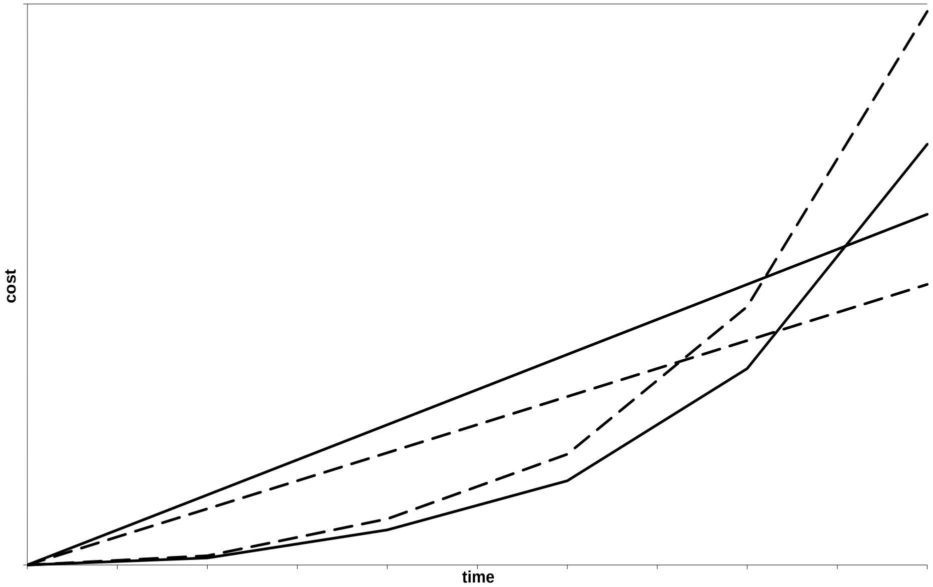


Figure 4.3: Original cost functions (solid lines) and scaled cost functions (dashed lines)

If $p < 2$ then $f_1(t) \geq f_2(t)$ for $0 \leq t \leq T_1$. This cannot be considered fair: Flight 2 has a larger interval and therefore more possible landing times. If it would be impossible to land both aircraft before time T_1, aircraft 2 is able to land between time T_1 and T_2, while aircraft 1 is not. When choosing $p < 2$ it is always cheaper to land aircraft 1 before aircraft 2 (even if both aircraft can be scheduled before time T_1) and airline 2 costs ($\kappa_2(t_2^*)$) will be larger than airline 1 costs ($\kappa_1(t_1^*)$). So when $p < 2$, airline 1 is better off, while airline 2 provides more flexibility.

If $p = 2$ then $f_1(t) = f_2(t)$ for $0 \leq t \leq T_1$, which can be considered fair. Another choice is to reward aircraft 2 for providing more flexibility (by a larger interval) by choosing $p > 2$. In our computational experiments, $p = 2$ is used.

In Figure 4.3 an example of the scaling of cost functions is depicted. This example considers two airlines that each have only one flight. The landing time interval for both flights is equal and spans the entire x-axis of the figure. One of the airlines uses a single linear function for its flight, while the other one uses a convex piecewise linear function with a lot of breakpoints. Both the original and scaled cost functions are shown in the figure. The area under the original linear cost function is much larger than the area under the original convex piecewise linear function. The application of the scaling mechanism results in an equal area under both cost functions.

The method can also be explained in the following way: Each airline “receives” a certain budget of scaled costs. The airline decides how to “spent” this budget. The budget has to be rationed over all flights of the airline and for each flight over the times in its landing time interval. This should be done by using a convex piecewise linear cost function for each flight representing the scaled cost at each possible landing time. The budgets for the airlines are determined such that the average amount of budget available per flight per possible landing time is equal for all airlines.

At most airports, there are peak periods during the day. If the α_a’s were determined for a day or longer, an airline with flights scheduled in peak and non-peak periods has an advantage over an airline with flights only scheduled in peak periods. The first airline could assign large cost to delays for their flights during peak periods, compared to their flights outside peak periods. The average scaled delay cost per time unit will be much larger for the first airline compared to the second for their respective flights in the peak periods. Therefore, a minimization of total scaled cost will lead to less delay for flights from airline 1 in peak periods. Runway queuing delays are much more likely to take place in peak periods, because in these periods the demand is close to (or even temporarily exceeds) capacity. Outside peak periods, queuing delays do not occur frequently, so flights from airline 1 are expected to receive little delay, again. Therefore, it is recommended to determine distinct scaling factors for separate (classes of) time intervals(such as peak and non-peak periods). This can be done by calculating distinct scaling factors for all flights (originally) scheduled per fixed length time period (e.g., every hour).

4.2 Fairness Definitions

Although everybody has a general idea about fairness, it is hard to give a formal definition, especially related to the aircraft landing problem.

The issues of fairness and equity (mainly from an economic viewpoint) are extensively discussed in Young [78]. Fairness is also considered in relation to some O.R. problems, such as game theory [15], bandwidth sharing problems in computer and telecom networks [54], queuing [11] and job scheduling [65]. However, it is difficult to apply these (problem-specific) definitions directly to the aircraft landing problem.

In this section we will give three fairness definitions related to the aircraft landing problem. These definitions will be used to formulate associated measures and optimization objectives.

Absolute Fairness

The most natural way to define fairness is an equal division (proportional to the number of flights) of scaled cost over the airlines. This means that we will compare the average scaled cost per flight of the airlines:

$$\bar{c}_a(t_1, \ldots, t_N) := \frac{1}{|F_a|} \sum_{i \in F_a} f_i(t_i) \qquad a \in A, \tag{4.3}$$

given a schedule where flight i lands at time t_i.

These average scaled cost per flight should be equal or almost the same for all airlines. To measure how fair a schedule is we can use the root mean square deviation:

$$\sigma_{\bar{c}}(t_1, \ldots, t_N) := \sqrt{\frac{1}{|A|} \sum_{a \in A} \left(\bar{c}_a(t_1, \ldots, t_N) - \frac{1}{|A|} \sum_{b \in A} \bar{c}_b(t_1, \ldots, t_N) \right)^2} \tag{4.4}$$

A problem is that the total scaled cost are not fixed but depend on the order and times the flights are scheduled. During the optimization these will change. This means that there will always be a trade-off between efficiency (total scaled cost) and fairness (the division of these total cost over the airlines).

Therefore, it would make no sense to change a schedule when the resulting schedule is fairer but leads to a cost increase for a single airline and no change in cost for all other airlines. On the other hand we do not want to lower the cost of an airline that has already low average cost at the expense of an airline with high average cost.

It is also hard to use the above measure directly in the optimization process. Therefore, during the optimization process the scaled cost of the airline that is the worst off (in the currently considered schedule) will be minimized:

$$\min_{t_1, \ldots, t_N} \max_{a \in A} \left\{ \bar{c}_a(t_1, \ldots, t_N) \right\} \tag{4.5}$$

Relative Fairness

It is natural to compare the current schedule to a reference schedule. Usually this will be an earlier schedule. Fairness can be assessed by comparing the airlines improvement compared to this reference schedule. This reference schedule has to be a feasible schedule (with respect to the current separation requirements). In this research the (feasible) schedule obtained by landing the flights in the order of their scheduled times as listed in the Original Airline Guide (OAG) schedule is used as reference schedule. This is the schedule that would be obtained without actively sequencing the flights and no disruptions from the original timetable occurring.

Let $\hat{t}_i$ be the landing time of flight i in the reference schedule. In this case we will compare the ratio of the airline cost in the considered schedule and the reference schedule:

$$\Delta_a(t_1, \ldots, t_N) := \frac{\sum_{i \in F_a} f_i(t_i)}{\sum_{i \in F_a} f_i(\hat{t}_i)} \qquad a \in A. \tag{4.6}$$

Ideally we want every airline to have some minimal improvement (or maximum deterioration) in the new schedule. Therefore we can measure this by the percentage of airlines that are worse off than in the reference schedule:

$$\frac{1}{|A|} \sum_{a \in A} 1_{\{\Delta_a(t_1, \ldots, t_N) > 1\}} \tag{4.7}$$

Equivalently, this can be measured by the percentage of airlines that have an improvement less than a certain fixed ratio.

Considering the trade-off between total cost and fairness, we can obtain this by maximizing the improvement of the airline that is worst off (lowest improvement) during the optimization process:

$$\min_{t_1, \ldots, t_N} \max_{a \in A} \left\{ \Delta_a(t_1, \ldots, t_N) \right\} \tag{4.8}$$

Fairness Measured by Delay

A trade-off between overall efficiency and fairness has to be found. A better trade-off might be found by using different measures for overall efficiency and fairness. This leads to the idea to use delay as measure for fairness.

To measure fairness by delay we will compare the average airline delay per flight compared to the reference schedule:

$$\bar{d}_a(t_1,\ldots,t_N) := \frac{1}{|F_a|}\sum_{i\in F_a}\left(t_i-\hat{t}_i\right)^+ \qquad a\in A. \tag{4.9}$$

Similarly as with the scaled cost, the fairness measured by delay can be evaluated by the root mean square deviation of the average airline delays:

$$\sigma_{\bar{d}}(t_1,\ldots,t_N) := \sqrt{\frac{1}{|A|}\sum_{a\in A}\left(\bar{d}_a(t_1,\ldots,t_N)-\frac{1}{|A|}\sum_{b\in A}\bar{d}_b(t_1,\ldots,t_N)\right)^2} \tag{4.10}$$

When considering fairness measured by delay in the optimization process we will minimize the average delay of the airline that is the worst off:

$$\min_{t_1,\ldots,t_N}\max_{a\in A}\left\{\bar{d}_a(t_1,\ldots,t_N)\right\} \tag{4.11}$$

4.2.1 Example

To illustrate the difference between our definitions of fairness and the trade-off with total cost, a small example is presented in this section.

The flights listed in Table 4.1 have to be scheduled for landing. We assume that the flights cannot land earlier than their timetable time. Between all flights a separation of 2 minutes is required. We assume that the cost are linear in the amount of delay. The cost per minute of delay differs per flight and is listed in the table. Note that the average cost per minute delay are the same for both airlines, so scaling is not necessary.

In Table 4.2 optimal schedules considering different objectives are shown. In Table 4.3 the delay and costs and their division over the airlines are shown.

Flight	Airline	Delay Cost	Timetable
1	A	2	00:00
2	A	6	00:01
3	B	2	00:02
4	B	4	00:03
5	A	5	00:04
6	B	7	00:05

Table 4.1: Flights to schedule

			Minimum	Fairness		
Flight	Timetable	FCFS	Cost	Absolute	Relative	Delay
1 (A)	00:00	00:00	00:00	00:00	00:00	00:00
2 (A)	00:01	00:02	00:02	00:02	00:02	00:02
3 (B)	00:02	00:04	00:10	00:10	00:10	00:08
4 (B)	00:03	00:06	00:08	00:04	00:04	00:04
5 (A)	00:04	00:08	00:04	00:08	00:06	00:10
6 (B)	00:05	00:10	00:06	00:06	00:08	00:06

Table 4.2: Schedules obtained by considering different objectives

The FCFS schedule is the minimum total cost schedule when the flights are ordered according to their timetable time. This schedule is used as reference schedule. In this case the total cost are 77 and the total delay is 15 minutes. Airline B has (almost) twice the delay and cost of airline A.

In the minimum total cost schedule, the total cost are only 49. However, almost all the delay and cost are at the expense of airline B.

If we focus on absolute fairness, the costs can be shared almost equally. The total cost are only slightly higher than in the minimum total cost schedule. Airline B has large savings compared to the FCFS schedule, while airline A's cost remain equal.

Considering relative fairness, it is possible to accomplish savings compared to the FCFS schedule for both airlines. Airline A saves 38% and airline B 20%. However, since the total cost are higher than in the minimum total cost schedule, there is a clear trade-off between relative fairness and total cost. In this particular case, the minimum cost schedule could be preferable when considering relative fairness, because this schedule gives

	Delay			Cost		
	A	B	Total	A	B	Total
FCFS	5	10	15	26	51	77
Min Cost	1	14	15	6	43	49
Absolute Fairness	5	10	15	26	27	53
Relative Fairness	3	12	15	16	41	57
Fairness by Delay	7	8	15	36	23	59

Table 4.3: Cost and delay for the airlines for the different schedules

almost the same savings for airline B (18%) and larger savings for airline A.

In the four schedules considered, airline B receives much more delay than airline A. When the objective is fairness measured by delay, we obtain a schedule with almost the same delay for both airlines. Again, there is a trade-off between total cost and fairness. This schedule has the highest total cost. However, Airline B has the lowest cost of all considered schedules. But airline A has more cost and delay than in any of the other schedules.

This example shows that the minimum total cost schedule can have an unequal spread of cost and delay over the airlines. By considering fairness explicitly, we can obtain schedules with a fairer spread of cost and delay. However, there is a trade-off because the total cost in these schedules are often larger. Schedules with different trade-offs between total cost, delay and fairness can be obtained.

Chapter 5

Using Airline Cost in the Aircraft Landing Problem

In this chapter the use of airline cost in the aircraft landing problem is considered. This is done using the approach to consider airline cost in a fair manner that was introduced in the previous chapter.

The single runway aircraft landing problem considers the scheduling of landings for a set of flights at a runway. The landing times are constrained to be within predefined time windows and to allow for the required separation between the flights. A literature overview of the aircraft landing problem can be found in Section 3.1.

In Section 5.1 a mathematical programming formulation of the problem is given. A problem specific local-search heuristic to efficiently obtain reasonable schedules is introduced in Section 5.2.

A large number of problem instances, created from real-life data concerning arrivals during a week at a major European hub, were used as input for our method in a static manner. The results are shown in Section 5.3. These results include an analysis of the costs and fairness of the obtained schedules as well as the performance of the heuristic (both the quality of the solutions and computation times). Next to this, a simulation study in which the model is used in a dynamic manner is presented in this section.

The thesis ends in Section 5.4 with a number of conclusions.

5.1 Mathematical Programming Formulation

In this section a Mixed Integer Programming (MIP) formulation of the single runway aircraft landing problem is given. The basic notation and constraints are similar to those of Beasley et al. [17].

5.1.1 Basic Notation

Let $F = \{1, \dots, N\}$ be the set of all flights to schedule. Let

E_i :	Earliest possible landing time for flight i	$i \in F$
L_i :	Latest possible landing time for flight i	$i \in F$
S_{ij} :	Required separation time when flight i lands before flight j	$i, j \in F, i \neq j$

The main decision variables are the landing times of the flights. The formulation requires some additional decision variables to represent the sequence of the flights:

t_i	: landing time of flight i	$i \in F$
δ_{ij}	$= \begin{cases} 1 & \text{if flight } i \text{ lands before flight } j \\ 0 & \text{otherwise} \end{cases}$	$i, j \in F, i \neq j$

5.1.2 Constraints

The first constraint ensures that the landing time falls in the possible landing time interval:

$$E_i \leq t_i \leq L_i \qquad i \in F \tag{5.1}$$

Constraint (5.2) ensures that either flight i lands before flight j or the reverse:

$$\delta_{ij} + \delta_{ji} = 1 \qquad i, j \in F, j > i \tag{5.2}$$

This ordering of the flights is needed to ensure the proper separation between flights. To obtain this, we introduce the following sets of ordered pairs of flights, determined by their possible landing time intervals :

U : the set of pairs (i, j) of flights for which it is undetermined whether flight i lands before flight j or the other way around;

V : the set of pairs (i, j) of flights for which flight i definitely lands before flight j, but for which the separation is not automatically satisfied;

W : the set of pairs (i, j) of flights for which flight i definitely lands before flight j, and the separation is automatically satisfied.

More formally:

$$\begin{aligned}
U &= \{(i,j)| & & E_j \leq E_i \leq L_j \text{ or } E_j \leq L_i \leq L_j \text{ or } E_i \leq E_j \leq L_i \text{ or} \\
& & & E_i \leq L_j \leq L_i, i, j \in F, i \neq j, \} \\
V &= \{(i,j)| & & L_i < E_j \text{ and } L_i + S_{ij} > E_j, i, j \in F, i \neq j\} \\
W &= \{(i,j)| & & L_i < E_j \text{ and } L_i + S_{ij} \leq E_j, i, j \in F, i \neq j\}.
\end{aligned}$$

In Figure 5.1 these sets are shown visually. The rectangles depict the possible landing time intervals of flights. The pairs $(1, 2)$ and $(2, 1)$ are in

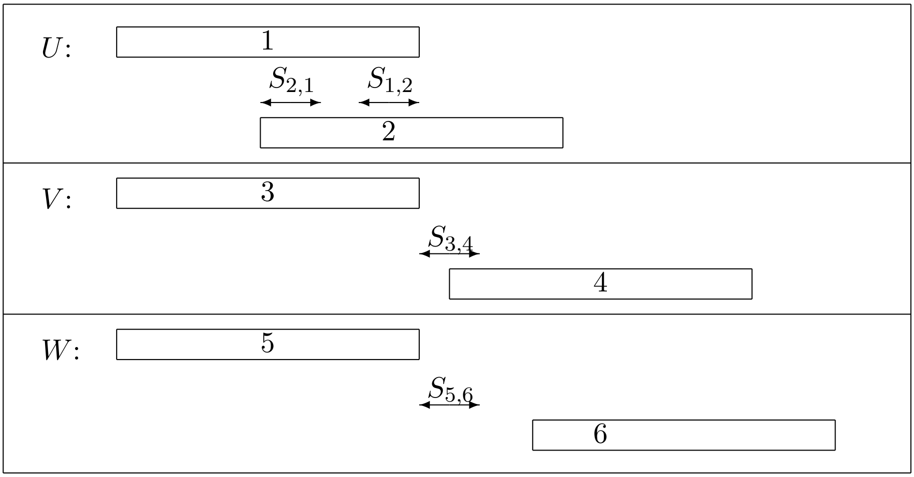


Figure 5.1: Sets U, V and W

set U. Pair $(3, 4)$ is in set V and pair $(5, 6)$ belongs to set W. Note that the pairs $(4, 3)$ and $(6, 5)$ are in none of the sets.

Using these sets we can formulate the following constraints to ensure the proper separation:

$$\delta_{ij} = 1 \qquad (i,j) \in V \cup W \tag{5.3}$$

$$t_j \geq t_i + S_{ij} \qquad (i,j) \in V \tag{5.4}$$

$$t_j \geq t_i + S_{ij}\delta_{ij} - (L_i - E_j)\delta_{ji} \qquad (i,j) \in U \tag{5.5}$$

If flight i definitely precedes flight j then we can fix δ_{ij} (constraint (5.3)). For $(i, j) \in V$ the order is known but the separation still needs to be ensured (constraint (5.4)). This must also be done for the pairs in U. This is done by constraint (5.5) for the pair (i, j) if flight i lands before flight j ($\delta_{ij} = 1, \delta_{ji} = 0$). If this is not the case, this constraint is superfluous. Note that if $(i, j) \in U$ then $(j, i) \in U$ and constraint (5.5) ensures the separation for both orders.

5.1.3 Cost Functions

In this section, we will work out how to express the convex piecewise linear cost functions in terms of our mathematical programming formulation.

As explained in Section 4.1 each airline provides a convex piecewise linear cost function for each of its flights. This cost function is then scaled to obtain equity among airlines. Let $f(x)$ be such a scaled convex piecewise linear cost function. The flight index is omitted here for readability. Let K be the number of breakpoints of the function. $f(x)$ can be written as a set of $(K + 1)$ linear functions with slopes $A_0, \ldots, A_K$ and intercepts $B_0, \ldots, B_K$:

$$f(x) = \begin{cases} A_0 x + B_0 & 0 \leq x \leq X_1 \\ A_1 x + B_1 & X_1 \leq x \leq X_2 \\ \vdots & \vdots \\ A_K x + B_K & X_K \leq x \end{cases} \tag{5.6}$$

An example of such a function with our notation depicted in Figure 5.2.

Theorem 5.1. *The function $f(x)$ can also be written in the following way:*

$$f(x) = \max_{k=0,\ldots,K} \{A_k x + B_k\}$$

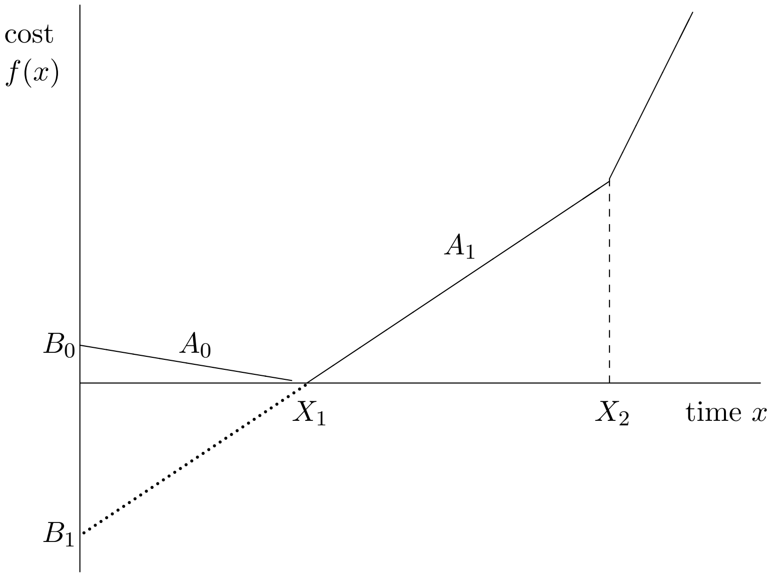


Figure 5.2: Example of a convex piecewise linear cost function

Proof. This follows from the continuity of $f(x)$ (in the breakpoints), implying

$$f(X_k) = A_{k-1}X_k + B_{k-1} = A_kX_k + B_k \qquad k = 1, \ldots, K \tag{5.7}$$

Next we will consider the function around a breakpoint X_k. We consider the points $X_k - \epsilon > X_{k-1}$ and $X_k + \epsilon < X_{k+1}$ with $\epsilon > 0$. Because the function is convex and by using (5.7) we obtain the following:

$$\begin{aligned}
f(X_k) &\leq \frac{1}{2}f(X_k - \epsilon) + \frac{1}{2}f(X_k + \epsilon) \\
A_{k-1}X_k + B_{k-1} &\leq \frac{1}{2}\Big(A_{k-1}(X_k - \epsilon) + B_{k-1} + A_k(X_k + \epsilon) + B_k\Big) \\
A_{k-1}X_k + B_{k-1} &\leq \frac{1}{2}\Big(A_{k-1}X_k - \epsilon A_{k-1} + B_{k-1} + \epsilon A_k + A_kX_k + B_k\Big) \\
A_{k-1}X_k + B_{k-1} &\leq \frac{1}{2}\Big(A_{k-1}X_k - \epsilon A_{k-1} + B_{k-1} + \epsilon A_k + A_{k-1}X_k + B_{k-1}\Big) \\
A_{k-1}X_k + B_{k-1} &\leq A_{k-1}X_k + B_{k-1} + \frac{1}{2}\epsilon(A_k - A_{k-1}) \\
A_{k-1} &\leq A_k
\end{aligned}$$

This holds for every breakpoint and thus

$$A_0 < A_1 < \ldots < A_K. \tag{5.8}$$

Using (5.8) and the continuity (5.7), the following holds for a point $x \in [X_k, X_{k+1}]$:

$$\begin{aligned} f(x) &= A_k x + B_k \\ &= A_k X_k + B_k + A_k(x - X_k) \\ &= A_{k-1} X_k + B_{k-1} + A_k(x - X_k) \\ &\geq A_{k-1} X_k + B_{k-1} + A_{k-1}(x - X_k) \\ &= A_{k-1} x + B_{k-1} \end{aligned}$$

Continuing in the same manner gives:

$$A_k x + B_k \geq A_{k-1} x + B_{k-1} \geq A_{k-2} x + B_{k-2} \geq \ldots \geq A_0 x + B_0 \tag{5.9}$$

Similarly,

$$\begin{aligned} f(x) &= A_k x + B_k \\ &= A_k X_{k+1} + B_k - A_k(X_{k+1} - x) \\ &= A_{k+1} X_{k+1} + B_{k+1} - A_k(X_{k+1} - x) \\ &\geq A_{k+1} X_{k+1} + B_{k+1} - A_{k+1}(X_{k+1} - x) \\ &= A_{k+1} x + B_{k+1} \end{aligned}$$

Continuing in the same manner gives:

$$A_k x + B_k \geq A_{k+1} x + B_{k+1} \geq A_{k+2} x + B_{k+2} \geq \ldots \geq A_K x + B_K \tag{5.10}$$

Combining equations (5.9) and (5.10) gives

$$f(x) = \max_{k=0,\ldots,K} \{A_k x + B_k\} \qquad \square$$

Comparing to the representation of a cost function as given by in equation (5.6), an additional subscript i is needed to indicate the flight. Consider the objective to minimize the total scaled cost

$$\min \sum_{i \in F} f_i(t_i).$$

The cost functions f_i $(i \in F)$ are not linear in the current decision variables t_i, and therefore additional decision variables c_i are necessary:

c_i : cost for landing flight i, $i = 1, \ldots, N$

c_i represents the cost function $f_i(t_i)$. To ensure this, the following constraints are introduced:

$$c_i \geq A_{ik} t_i + B_{ik} \qquad i = 1, \ldots, N; k = 0, \ldots, K_i \tag{5.11}$$

These ensure for flight i that

$$c_i \geq \max_{k=0,\ldots,K_i} \{A_{ik} t_i + B_{ik}\} = f_i(t_i).$$

The minimum scaled cost objective can now be written in the following manner:

$$z = \min \sum_{i=1}^{N} c_i \tag{5.12}$$

Equations (5.11) and (5.12) ensure that $c_i = f_i(t_i)$:

Suppose $t_1^*, t_2^*, \ldots, t_n^*, c_1^*, c_2^*, \ldots, c_n^*$, is an optimal solution of our MIP, where: $c_i^* > f_i(t_i^*)$ for some i. Let $c_i' = f_i(t_i^*)$. Replacing c_i^* with c_i' will decrease the objective by $c_i^* - f_i(t_i^*)$ without violating any of the constraints (5.11):

$$c_i' = f_i(t_i^*) = \max_{k'=0,\ldots,K_i} \{A_{ik'} t_i^* + B_{ik'}\} \geq A_{ik} t_i^* + B_{ik}$$

for $k = 0, \ldots, K_i$. So $c_i^* > f_i(t_i^*)$ cannot be optimal (and $c_i' = f_i(t_i^*)$ is).

5.1.4 Fairness Objectives

In this section, we will introduce alternative objectives and additional constraints which are related to our fairness definitions, as presented in Section 4.2. One must realize that there is a trade-off between the total (scaled) cost and fairness. The additional constraints that are necessary to ensure the fairness criteria are met can cause additional costs.

Absolute Fairness

The objective to minimize the maximum average airline cost per flight as defined in equation (4.5) is easily formulated in terms of our MIP model.

Let $c_{\max}$ be the decision variable that represents the maximum airline average cost. We can model this using the following objective and additional constraints.

$$\min c_{\max} \tag{5.13}$$

$$c_{\max} \geq \frac{1}{|F_a|} \sum_{i \in F_a} c_i \qquad \forall a \in A \tag{5.14}$$

As we mentioned before there is always a trade-off between total cost and fairness, therefore we adapt the objective to represent this trade-off:

$$\min c_{\max} + \frac{\epsilon}{|F|} \sum_{i \in F} c_i \tag{5.15}$$

ϵ should be chosen small ($0 < \epsilon \ll 1$) to focus on absolute fairness. By doing this, among the solutions with minimum $c_{\max}$ the solution with the smallest total cost will be obtained. By choosing $\epsilon = 1$, reducing the overall average cost per flight is considered equally important as reducing the average cost per flight for the airline with the largest cost.

Relative Fairness

Equation (4.8) can be formulated in terms of the MIP model as follows.

Let $\hat{c}_i$ be the cost involved with landing flight i using the reference schedule. Let $\Delta_{\max}$ be the maximum airline ratio of the cost in the current schedule and the reference schedule. So $(1 - \Delta_{\max})$ represents the minimum improvement. We can now model this using the following objective and constraints.

$$\min \Delta_{\max} \tag{5.16}$$

$$\Delta_{\max} \geq \frac{\sum_{i \in F_a} c_i}{\sum_{i \in F_a} \hat{c}_i} \qquad \forall a \in A \tag{5.17}$$

Again we want also to consider the trade-off between total cost and fairness, by adapting the objective in the following way:

$$\min \Delta_{\max} + \epsilon \frac{\sum_{i \in F} c_i}{\sum_{i \in F} \hat{c}_i} \tag{5.18}$$

Fairness measured by delay

A trade-off between overall efficiency and fairness has to be found. A better trade-off might be found by using different measures for overall efficiency and fairness. This leads to the idea to use airline delay (instead of costs) as measure for fairness. The minimum of the maximum average airline delay per flight (equation (4.11)) is used as fairness objective. Let $\hat{t}_i$ be the timetable landing time of flight i. Decision variable d_i represents the delay of flight i and $d_{\max}$ represents the maximum airline average delay.

We can now model the trade-off between maximum airline average delay and total cost using the following objective and constraints:

$$\begin{aligned}
\min d_{\max} &+ \frac{\epsilon}{|F|} \sum_{i \in F} c_i && (5.19)\\
d_{\max} &\geq \frac{1}{|F_a|} \sum_{i \in F_a} d_i \qquad \forall a \in A && (5.20)\\
d_i &\geq t_i - \hat{t}_i \qquad \forall i && (5.21)\\
d_i &\geq 0 \qquad \forall i && (5.22)
\end{aligned}$$

5.2 Local Search Heuristic

As discussed in Section 3.1, the complexity of the problem makes it computationally difficult to find optimal solutions for realistically sized instances. Therefore a heuristic is needed to find good solutions within a reasonable time.

The idea behind the local search heuristics proposed in this research is to repeatedly find an improved sequence for the flights and determine the optimal landing times given this sequence. This idea stems from the following observation: If the landing sequence of the flights is given, the MIP-formulation becomes an LP formulation, since the values of all the binary variables are known. This formulation consists of the constraints (5.1), (5.5) and (5.11), combined with one of the objective functions and the corresponding constraints. The solution of this LP provides the optimal landing times, given the sequence. LPs can be solved efficiently, i.e., in polynomial time.

A local search method can be used to find and improved sequence of the flights. Local search uses a neighborhood of the current solution to find a new (improved) solution. The neighborhood is defined in such a way that new sequences will be “close” to the current sequence, meaning they are very similar. This means the corresponding LP formulations will also be. Most LP solvers are able to solve such a formulation very efficiently by using the solution obtained for the previous neighbor.

The general local search algorithm is given below.

```
LOCAL SEARCH()
$S =$ initial feasible solution
while there is a neighbor of $S$ of better quality
    do $S =$ neighbor of $S$ of better quality
```

In Section 5.2.1 we will specify how to find an initial feasible solution. Two different neighborhoods are presented in Section 5.2.2. A selection procedure for a neighbor of better quality is presented in Section 5.2.3. There are standard techniques available to do this. However, it is beneficial to use problem specific features in these procedures.

5.2.1 Initial Feasible Solution

Note that the problem of finding a feasible solution is NP-complete in itself. This follows from the fact that the problem to minimize tardiness when scheduling jobs with release and due dates on a single machine is already NP-hard, as shown in [33].

Because of this complexity, we will use an approach that resembles the way flights are landed in practice. This involves landing the flights in the order they approach the airport. Therefore this order is called the First Come First Served (FCFS) sequence. When scheduling, this means the sequence is formed by the flights, sorted according to their expected arrival times. In current practice only small changes to this sequence are possible. If this does not lead to a feasible solution it is impossible to land the flights at the airport, using the current procedures. In this situation some of the flights are currently canceled or diverted to other airports. These kinds of decisions fall outside the scope of our model.

However, we will consider a wider range of sequences in order to find an initial feasible solution. These sequences are obtained by (repeatedly) applying operations on the current sequence until a feasible schedule is found. These operations alter the current sequence such that it will be more likely to obtain a feasible sequence. If this procedure still does not give a feasible solution, our heuristic cannot be used to find a solution. In our experiments (see Section 5.3) we were always able to find an initial feasible solution. Occasionally the procedure had to be performed because the FCFS sequence did not immediately give a feasible schedule.

Let us explain the procedure formally: Let π be a sequence of the N flights, and $\pi(i)$ the flight at position i in the sequence π. Now let $\hat{t}_i \in [E_i, L_i]$ be the timetable or expected landing time of flight i. The initial sequence π is obtained by sorting the flights in order of non-decreasing $\hat{t}_i$. In case of ties the flights (with equal $\hat{t}_i$'s) are ordered in non-decreasing order of E_i, ties broken arbitrarily. So the following holds:

$$\hat{t}_{\pi(1)} \leq \hat{t}_{\pi(2)} \leq \ldots \leq \hat{t}_{\pi(N)}$$

$$E_{\pi(i)} \leq E_{\pi(i+1)} \qquad \text{if } \hat{t}_{\pi(i)} = \hat{t}_{\pi(i+1)}$$

Solving the LP gives optimal feasible landing times for this (FCFS) sequence. These landing times form the First Come First Served (FCFS) schedule. It is, however, possible that the LP is not feasible, meaning there are no feasible landing times given this sequence. This means there is no schedule where all flights land in their possible landing interval (constraints (5.1)) and all separation regulations are met (constraints (5.5)). In that case new sequences are repeatedly generated, by swapping two adjacent flights for which the earlier one has a larger latest landing time, until the LP of such a sequence is feasible.

To be more precise in each iteration flight $\pi(i^*)$ and $\pi(i^*{+}1)$ are swapped, where

$$i^* = \arg \max_{i=1,\ldots,N-1} \{L_{\pi(i)} - L_{\pi(i+1)} : L_{\pi(i)} - L_{\pi(i+1)} \geq 0\}.$$

If exhaustively performing the above swaps still does not yield a feasible solution, separation between adjacent flights will be considered to obtain new sequences. Let $\hat{S}^{\pi}(i)$ be the decrease in separation between adjacent flights when swapping flight $\pi(i)$ and $\pi(i+1)$:

$$\begin{aligned}\hat{S}^{\pi}(i) &:= S_{\pi(i-1),\pi(i)} + S_{\pi(i),\pi(i+1)} + S_{\pi(i+1),\pi(i+2)} \\ &- S_{\pi(i-1),\pi(i+1)} - S_{\pi(i+1),\pi(i)} - S_{\pi(i),\pi(i+2)}\end{aligned}$$

with $S_{\pi(0),\pi(1)} = S_{\pi(N),\pi(N+1)} := 0$.

New sequences are repeatedly generated, by swapping two adjacent flights, for which this separation is decreased, i.e. swap flight $\pi(i^*)$ and $\pi(i^*+1)$ where

$$i^* = \arg \max_{i=1,\dots,N-1} \{\hat{S}^{\pi}(i) : \hat{S}^{\pi}(i) \geq 0\}.$$

The procedure to find an initial feasible solution can be summarized as follows:

```
FIND INITIAL SOLUTION()
$\pi =$ sequence of sorted flights
while $LP(\pi)$is not feasible
    do $\pi =$ sequence which is more likely to be feasible than $\pi$
```

5.2.2 Neighborhoods

We will define two types of neighborhoods for the local search heuristic.

The first is a problem-specific extension of a swap neighborhood. The swap neighborhood consists of all sequences that are equal except that two flights have swapped positions.

The second is a problem-specific extension of a shift-neighborhood. The shift neighborhood consists of all sequences that are equal except that one flight is removed from its original position and inserted at a new position.

The problem-specific extensions lead to larger search spaces and thus possibly to better solutions. The extensions are described more formally in the next two sections.

Swap Neighborhood

This neighborhood consists of all sequences that are equal except that two flights have swapped positions. Hence, each solution has at most $N(N-1)/2$ neighbors.

Swapping a pair of flights can lead to an infeasible sequence. This will certainly be the case if the possible landing time intervals of the two swapped flights do not overlap, e.g., swap flights $\pi(i)$ and $\pi(j)$ with $i < j$ and $L_{\pi(i)} < E_{\pi(j)}$. These swaps are excluded from the neighborhood.

Another swap that leads to an infeasible sequence is the case where the possible landing time interval of a flight positioned between the two swapped flights does not overlap with one of the intervals of the swapped flights. It is never feasible to alter the order of these flights. But the sequence can be adjusted to avoid this. This is done by changing the position of the non-overlapping flight, such that the sequence is feasible with respect to the landing time intervals. The adjusted sequence is included in the neighborhood. This sequence does not guarantee a feasible solution, because the separation constraints can still cause infeasibility. This is verified by solving the LP.

Let us make this more precise. Let π be the feasible sequence of the flights in the previous iteration, where $\pi(i)$ is the flight at position i of sequence π. Consider swapping the flights at position i and j $(i < j)$ in π as depicted in Figure 5.3. The rectangles depict the possible landing intervals of the flights. We can only swap flight i and j if $L_{\pi(i)} > E_{\pi(j)}$, meaning that flight $\pi(j)$ can land before flight $\pi(i)$. Suppose the flight at position k $(i < k < j)$, cannot land earlier than flight $\pi(i)$ $(E_{\pi(k)} > L_{\pi(i)})$ but can land earlier and later than flight $\pi(j)$ $(E_{\pi(j)} \leq E_{\pi(k)} \leq L_{\pi(j)})$. Now swapping the flights at positions i and j would cause infeasibility because flight $\pi(k)$ would be positioned before flight $\pi(i)$.

However, this swap can be made feasible (w.r.t. the landing time intervals) by moving flight $\pi(i)$ to position $j-1$, moving flight $\pi(k)$ to position j, moving flights $\pi(k+1), \pi(k+2), \ldots, \pi(j-1)$ to positions $k, k+1, \ldots, j-2$ and of course flight $\pi(j)$ to position i. This procedure can be performed

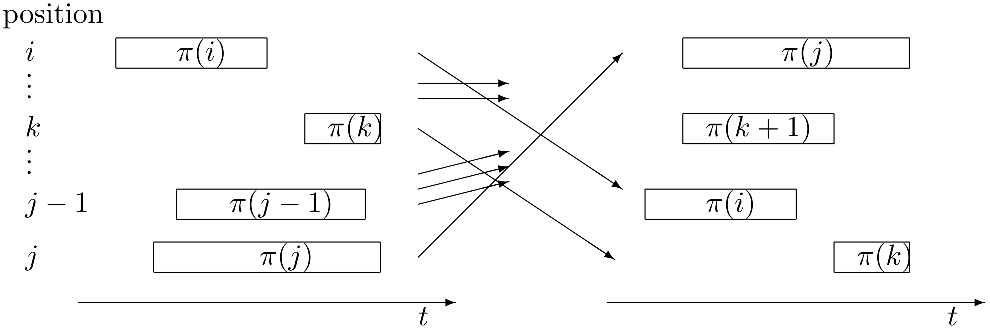


Figure 5.3: Swap neighborhood

repeatedly for all flights between positions i and j which cannot land before flight $\pi(i)$ and a similar procedure can be performed repeatedly for all flights between positions i and j which cannot land after flight $\pi(j)$.

We will explain this further with an example. Consider the following flights:

Flight	Possible landing time interval	
A	[0,	100]
B	[0,	150]
C	[120,	200]
D	[50,	200]
E	[0,	200]

Assume the current sequence is A,B,C,D,E and we want to swap flight A and E. However, no feasible landing times can be determined for the sequence E,B,C,D,A. This is caused by flight C, which cannot land before flight A, because its earliest landing time is after the latest landing time of flight A. However, if we move flight C together with flight A, the obtained sequence E,B,D,A,C might be feasible.

Shift Neighborhood

This neighborhood consists of all sequences that are equal except that one flight is removed from its original position and inserted at a new position. Hence, each solution has at most $N(N-1)$ neighbors.

Again, not all shifts lead to a feasible sequence. Consider the shift of the flight at position i to position $p > i$. This shift does not lead to a feasible solution if $E_{\pi(p)} > L_{\pi(i)}$. Similarly the shift of flight i to position $p < i$ does not lead to a feasible solution if $E_{\pi(i)} > L_{\pi(p)}$. These shifts are excluded from the neighborhood.

If there is a flight at a position between i and p that has a landing interval that does not overlap with the landing time interval of the flight at position i, the shift of the flight at position i to position p is not feasible, either. In this case the new sequence can be adjusted to obtain a sequence that is more likely to be feasible. This is done by changing the position of the non-overlapping flight, such that the sequence is feasible with respect to the landing time intervals. The adjusted sequence is included in the neighborhood. This sequence does not guarantee a feasible solution, because the separation constraints can still cause infeasibility. This is verified by

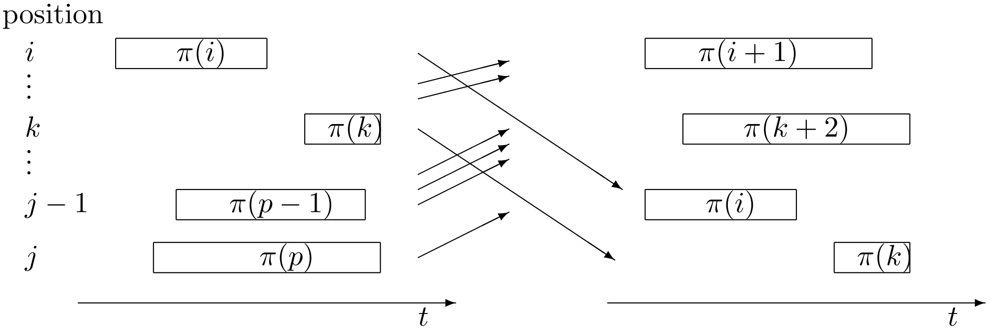


Figure 5.4: Shift neighborhood

solving the LP.

Let us make this more precise. For the sequence of flights as considered in the previous example, consider a shift that removes the flight at position i and inserts it at position $p > i$ (see Figure 5.4). Normally, flights $\pi(i+1), \ldots \pi(p)$ will be moved to positions $i, \ldots, j-1$, but since flight $\pi(k)$ cannot land before flight $\pi(i)$, we can only move flights $\pi(i+1), \ldots \pi(k-1)$ to positions $i, \ldots, k-2$. However, we can make this shift feasible by moving flight $\pi(k)$ to position p and flight $\pi(i)$ to position $p-1$. Further flights, $\pi(k+1), \ldots, \pi(p)$ are moved to positions $k-1, \ldots, p-2$.

Again this procedure can be performed repeatedly for all flights between positions i and j which cannot land before flight $\pi(i)$. A similar procedure is used for $p < i$.

We will explain this further with an example. Consider the following flights:

Flight	Possible landing time interval
A	[0, 100]
B	[0, 150]
C	[120, 200]
D	[50, 200]
E	[0, 200]

Assume the current sequence is A,B,C,D,E and we want to shift flight A to the 5th position. However, no feasible landing times can be determined for the sequence B,C,D,E,A. This is caused by flight C, which cannot land before flight A, because its earliest landing time is after the latest landing time of flight A. However, if we move flight C together with flight A, the obtained sequence B,D,E,A,C might be feasible.

5.2.3 Selection of a neighbor

The evaluation of a neighbor involves solving an LP. Although, each individual LP can be solved fast, the number of LPs to be solved can be considerable. Therefore, it is preferable to find a selection method that finds an improvement by evaluating as little neighbors as possible, to reduce the overall computation time.

Therefore, we will evaluate promising neighbors first. For all neighbors of the current solution, an estimated gain in objective is calculated. This estimation uses the landing times in the current solution, to estimate the landing times and involved scaled cost in the new solution, without solving the LP.

This estimation can be performed in several ways. The simplest approach is to estimate the new landing time for a flight, by the landing time from the flight at the same position in the previous solution and calculate the scaled cost. For example, the gain of swapping flights i and j that are scheduled to land at time t_i and t_j in the current solution, will be estimated by

$$f_i(t_i) + f_j(t_j) - f_i(t_j) - f_j(t_i).$$

The neighbors are evaluated in order of decreasing estimating gains: The neighbor with the largest estimated gain is evaluated first. If this neighbor indeed gives a better solution, it is selected. Otherwise the neighbor with the second largest estimated gain is evaluated, etc.

5.2.4 Fairness Neighborhood Restrictions

When minimizing total scaled cost, the flights can be considered independent of their airlines. Any cost improvement by any flight helps to achieve the objective. When using one of the fairness objectives that were introduced in Section 5.1.4, this is different. In this case, it is possible to restrict the neighborhoods by only allowing neighbors that involve flights of a subset of the airlines.

When absolute fairness is considered this could be the subset of airlines for which the average scaled cost in the current solution are close (enough) to the maximum airline average scaled cost. That is, airlines $a \in A$ for which

$$\frac{1}{|F_a|}\sum_{i \in F_a} c_i \geq \beta \max_{a \in A}\left\{\frac{1}{|F_a|}\sum_{i \in F_a} c_i\right\} \qquad 0 \leq \beta \leq 1. \tag{5.23}$$

β represents an airline selection threshold which determines how sensitive this control is. Choosing $\beta = 1$ will only allow neighbors that involve flights of the maximum cost airline(s). This might be too restrictive and might lead to bad local minima (w.r.t. total scaled cost). The same can be said of β close to 0 w.r.t. fairness criteria. β can be dynamically updated during the algorithm. Thus, first a solution that has little total cost can be found and than the fairness of this solution can be improved. This can be done by choosing $\beta = 0$ initially and increasing β with small steps with each improvement in objective.

When relative fairness is considered a subset of airlines for which the cost ratio in the current solution are close enough to the maximum cost ratio can be considered:

$$\frac{\sum_{i \in F_a} c_i}{\sum_{i \in F_a} \hat{c}_i} \geq \beta \max_{a \in A}\left\{\frac{\sum_{i \in F_a} c_i}{\sum_{f \in F_a} \hat{c}_i}\right\} \qquad 0 \leq \beta \leq 1. \tag{5.24}$$

5.2.5 Summary

The complete local search algorithm can be summarized as follows:

```
LOCAL SEARCH()
$\pi$ = FIND INITIAL SOLUTION()
$N(\pi)$ := Set of neighbors of $\pi$
Estimate gains for all members of $N(\pi)$
while $N(\pi) \neq \emptyset$
    do $\pi'$ = neighbor with maximum estimated gain in $N(\pi)$
        if $LP(\pi')$ is feasible and $z_{LP(\pi')} \leq z_{LP(\pi)}$
          then $\pi = \pi'$
                $N(\pi)$ = Set of neighbors of $\pi$
                Estimate gains for all members of $N(\pi)$
          else $N(\pi) = N(\pi) \setminus \pi'$
return $\pi$ and the optimal landing times for this sequence
```

5.3 Computational Experiments

Computational experiments were performed to assess the impact of considering airline cost functions in the aircraft landing problem and the related fairness issues. Flight schedule data from a major European hub were used to perform these experiments. In Section 5.3.1 more details about the data is provided.

In order to evaluate the performance of the local search heuristics, experiments were performed comparing heuristic solutions to optimal solutions. These results are presented in Section 5.3.2.

A large number of instances, created using the airport data, were tested using different variants of the local search heuristic. A detailed discussion of the results using various criteria to consider airline cost, delay and fairness, is provided in Section 5.3.3.

These experiments were performed in a static manner. This means we solve every instance once considering the situation at a single point in time (several hours before the actual arrivals). In practice, unexpected departure delays and weather changes occur during operations. Consequently, possible landing intervals and required separation times may change (after solving the problem). Our formulation and heuristic allow for a dynamic use of the model by recalculating the schedule, every time the circumstances change. It is interesting to evaluate what the impact of rescheduling on the cost savings is. A simulation study was conducted to assess this and the results are presented in Section 5.3.4.

5.3.1 Airport Data

The data contain all arrivals from a week in September 2004 at a large European hub. The data contain 3978 flights of 121 different airlines. The data include airline, flight number, aircraft type, arrival runway and scheduled and actual arrival times.

It is assumed that the cost function of every flight has a minimum of zero cost at the scheduled time of arrival of the flight, according to the timetable.

The structure of the cost functions for flights of the home carrier and its partners was determined in cooperation with specialists from this airline. Its perceived delay costs are strongly related to the number of missed transfers. This is quite natural, since this airline uses the airport as hub, and consequently has a lot of passengers transferring at the airport. Exact passenger flows and related costs were not provided by the airline for reasons of confidentiality. Instead the (incremental) numbers of missed transfers per 15 minutes of delay per flight were obtained from a poisson distribution (with a time interval dependent parameter). These were translated into convex piecewise linear cost functions by using the cumulative number of missed transfers up to time t as the slope of the cost function at time t.

Since the other airlines have a much smaller number of flights landing at the airport and therefore hardly any transfer passengers, it is assumed that their costs are mainly determined by punctuality. The (marginal) cost of delay will still vary between flights. The structure of the cost functions is however similar: Arriving earlier than the preferred landing time incurs relatively small marginal cost. After the preferred landing time larger marginal delay cost are assumed. A time after which marginal costs further increase is also defined (e.g., representing a time at which delays will propagate to other flights). The exact time this is occurring and the different marginal costs during the different time periods for each flight are obtained from probability distributions.

Our planning horizon is several hours before the flights will land. That means that flights from within Europe have not departed when planning. These flights are assumed to have a maximum departure delay and a possible landing interval of 3 hours. Intercontinental flights are en-route and are able to arrive between 25 minutes before and 30 minutes after schedule. This might seem a little optimistic, but is done in order to be able to assess the potential savings that can be obtained by (starting) scheduling relatively long in advance.

The required separation *time* between two flights can be calculated using the weight categories and approach speeds of the aircraft used for the flight.

From the total dataset containing 3978 flights, 139 instances were created by dividing the arrivals by runway and time. A runway is only used continuously for at most a few hours, depending on demand and weather conditions. The flights landing in such a period on a runway, are considered as a single instance. These instances contain up to 117 flights. The cost scaling mechanism was applied for each instance separately.

To evaluate the results, the schedules obtained by the heuristics are compared to the FCFS schedule. This is the schedule that is obtained without actively sequencing the flights. It is assumed that the flights land (respecting the required separation) in the order in which they approach the airport. As mentioned before this schedule resembles current practice. For our data it means that the FCFS order is determined by the scheduled times in the timetable. The landing times for the FCFS schedule are obtained by solving the LP given the FCFS order. Thus, these landing times respect the operational separation requirements and are within the feasible landing time intervals. The FCFS schedule is also used as initial solution by the local search heuristic (see Section 5.2.1).

5.3.2 Heuristic Performance

A first experiment has been performed to assess the quality of the solutions obtained by the local search heuristic. These were compared to the optimal solutions, obtained by CPLEX's MIP-solver. Because of the large computation times a limited number of instances (mostly smaller ones) were tested. These instances were tested using the minimum scaled cost objective (5.12) and both the swap and shift neighborhoods (without fairness neighborhood restrictions) as presented in Section 5.2.2.

This was done for 24 instances. One of these instances contained 117 flights, the others between 15 and 51 flights each. 19 instances were solved, assuming good visibility conditions. 5 instances were solved under low-visibility conditions, requiring a minimum radar separation of 6 nautical miles.

The total optimal scaled costs of all the instances, were 19% of the total scaled FCFS costs (81 % savings obtained) for these 24 instances. The heuristic using the swap and shift neighborhood, was able to achieve 94% and 98% of the total possible savings and gave the optimal solution in 5 and 13 instances, respectively. In the 5 instances with low-visibility conditions the swap and shift neighborhood achieved 90% and 85% of the total possible savings. This indicates that the heuristic provides solutions that obtain a large amount of the maximum possible savings. Note that the heuristic always provides a solution that is as least as good as the FCFS solution, since this is the initial solution used. The shift neighborhood seems to give better solutions under good visibility conditions and the swap neighborhood under low-visibility conditions. An intuitive explanation for this difference is that under low-visibility conditions more drastic changes are needed to improve the solution. No relation between solution quality and the size of the instances was found using both neighborhoods.

The computation time of the heuristic is mainly determined by the number of neighbor (LP) evaluations. In each iteration the neighbors of the

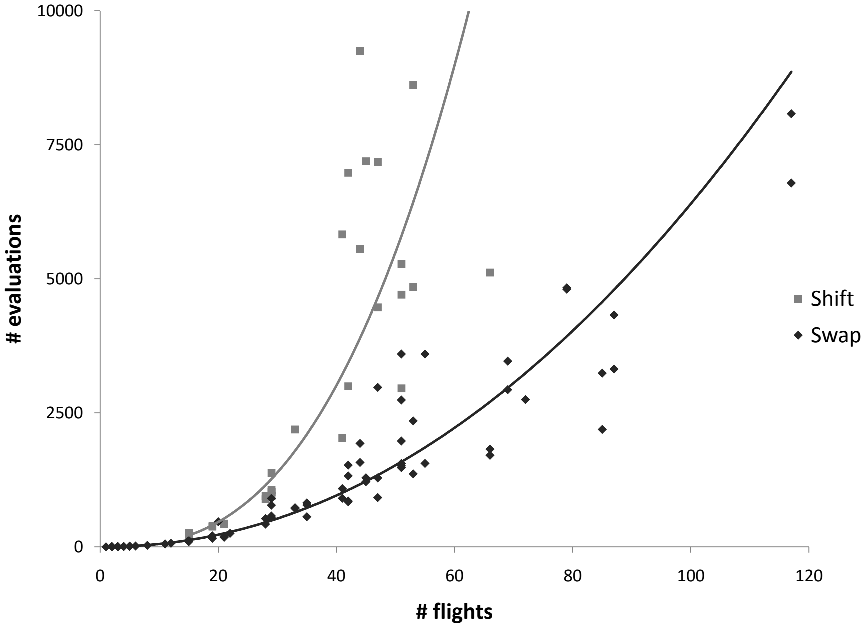


Figure 5.5: Number of neighborhood evaluations required using the shift and swap neighborhoods

current solution are evaluated until a better solution is found or all neighbors are evaluated. In the latter case the heuristic terminates. In Figure 5.5 the relation between the number of evaluations and the number of flights in the instance is shown. It is clear that the swap neighborhood requires less evaluations. This can be explained by the number of neighbors for a arbitrary sequence of N flights. The swap and shift neighborhood contain $\frac{N(N-1)}{2}$ and $N(N-1)$ neighbors respectively. This explains also that the shift neighborhood requires more iterations on average before a locally optimal solution is found. An evaluation takes approximately the same time for both neighborhoods, because a similar LP-formulation of the same size has to be solved. The time required to solve this LP increases somewhat with the number of flights in the instance.

The total running time of the heuristic, when using the swap neighborhood, is within a minute for instances with up to 40 flights and this increases to about ten minutes for instances with around 80 flights. Solving the MIP, gives large differences in computation times. Many of the smaller instances are solved within a minute. However, computation times of over 2.5 hours are also observed. The computations were performed on a Compaq computer with an Intel Pentium III processor (866 MHz), 256 MB physical memory and a Linux operating system. The heuristics are implemented in C++. The LP and MIP problems are solved using CPLEX.

In a practical setting a schedule might be needed fast. When running the local search heuristic, the best found solution so far is available at any time. This makes the local search heuristic extremely suitable for practical use.

The computation times can be reduced by limiting the size of the neighborhoods by allowing only swaps of flights that are less than a certain number of positions apart in the current sequence and shift a flight less than a certain number of positions. Another possibility is to skip the evaluation of neighbors with a low estimated gain. These changes might however affect the quality of the solutions.

5.3.3 Airline Cost and Fairness Experiments

In this section the computational experiments performed to assess the impact of using airline cost in the aircraft landing problem are discussed. Different variants of the local search heuristic aimed at finding different trade-offs between costs, delay and fairness were used in the experiments. Each variant was used to obtain a schedule for each of the instances. The resulting schedules are evaluated using different criteria considering airline cost, delay and fairness.

Heuristics

In Table 5.1, eight heuristics are listed, combining different objective functions, neighborhoods and neighborhood selection criteria. These heuristics are aimed at finding different trade-offs between cost, delay and fairness. The second column denotes the objective used. These objectives were introduced in sections 5.1.3 and 5.1.4. The third column indicates whether fairness neighborhood restrictions, as introduced in Section 5.2.4, are used.

Heuristic (1) is the reference schedule that resembles the current practice. This is the minimum cost schedule given that the flights will land in timetable order. To find this schedule a LP problem is solved.

Heuristic (2) is aimed at finding the minimum scaled cost schedule using the local search heuristic.

	Objective	Neighborhood Restrictions
(1)	total scaled cost (FCFS)	
(2)	total scaled cost	no
(3)	absolute fairness	yes
(4)	relative fairness	yes
(5)	absolute fairness	adaptive threshold
(6)	absolute fairness	after cost minimization
(7)	fairness measured by delay (scaled cost)	no
(8)	fairness measured by delay (real cost)	no

Table 5.1: Heuristics

Heuristic (3) uses the absolute fairness objective (5.15). The neighborhood is restricted by airline selection (with a fixed threshold $\beta = 0.9$).

Heuristic (4) uses the relative fairness objective (5.18). The neighborhood is restricted by airline selection (with a fixed threshold $\beta = 0.9$).

Heuristic (5) uses the absolute fairness objective (5.15) again. The neighborhood is restricted by airline selection with an adaptive threshold starting at 0 and increasing with 0.1 with every improvement of the objective until a maximum value of 0.9.

Heuristic (6) uses the absolute fairness objective (5.15) again. To avoid fair solutions with large total cost, the solution from heuristic (2) is used as initial solution. The neighborhood is restricted by airline selection (with a fixed threshold $\beta = 0.9$).

Both heuristic (7) and (8) measures fairness by delay using (LP) objective (5.19). The neighborhood is not restricted by airline selection. Heuristic (7) uses the total scaled cost as the efficiency measure. Heuristic (8) uses the original cost functions as supplied by the airlines. The cost scaling is not performed in this case because the fairness is measured by delay.

When the fairness objectives (5.15), (5.18) or (5.19) are used, $\epsilon = 0.01$ is used. Thus, fairness is considered as prevailing objective and the results will give an indication of the maximum fairness that can be achieved.

The local search heuristics (2)-(8) can be used with a shift and swap and neighborhood. However, when using fairness neighborhood restrictions it is more natural to use a shift neighborhood. This is because in a shift operation one flight (from a single airline) at a time is considered, whereas in a swap operation two flights (possibly from different airlines) are considered.

Results

Each of the eight heuristics is used to obtain a schedule for each of the European hub instances. The heuristics are implemented in C++. The LP problems are solved using the COIN-OR LP solver [1].

In the experiments, low-visibility conditions requiring 6 nautical miles separation were used. Although this leads to an equal separation distance required between every pair of flight, the *separation times* still differ because of the difference in approach speeds. This separation causes a decreased arrival capacity which will often result in large delays. It is interesting to assess the (scaled) costs and their distribution over the airlines, resulting from those delays.

The minimum total scaled cost heuristic (2) was tested using both the shift and swap neighborhood. Both neighborhoods gave tremendous cost savings compared to the FCFS schedule, which resembles the current practice. There was little difference between the neighborhoods. The overall cost savings were 33% of the costs (47% in terms of scaled costs) for the swap neighborhood and 35% of the costs (42% in terms of scaled costs) for the shift neighborhood.

Because of the small differences of the neighborhoods, heuristics (3)-(8) were only tested using the shift neighborhood. This neighborhood is also a natural choice when using fairness neighborhood restrictions. In the remainder of this section we will use the shift neighborhood results for heuristic (2).

In Figure 5.6 the average cost per flight using heuristic (2) for the 10 airlines with the largest total number of flights in the instances are shown.

It can be seen that these cost differ a lot between the airlines. This is already the case for the FCFS schedule. This means that the costs of the airlines are partly determined by the original timetable. Flights in peak periods are more likely to receive a delay and the corresponding costs. Although all of these airlines are better off than in the FCFS schedule, which is a validation for our scaling method, there are considerable differences between the (absolute and relative) savings of these airlines. Below we will assess whether the heuristics that are focused on fairness will decrease these differences. These results will be presented relative to the FCFS schedule, resulting from heuristic (1).

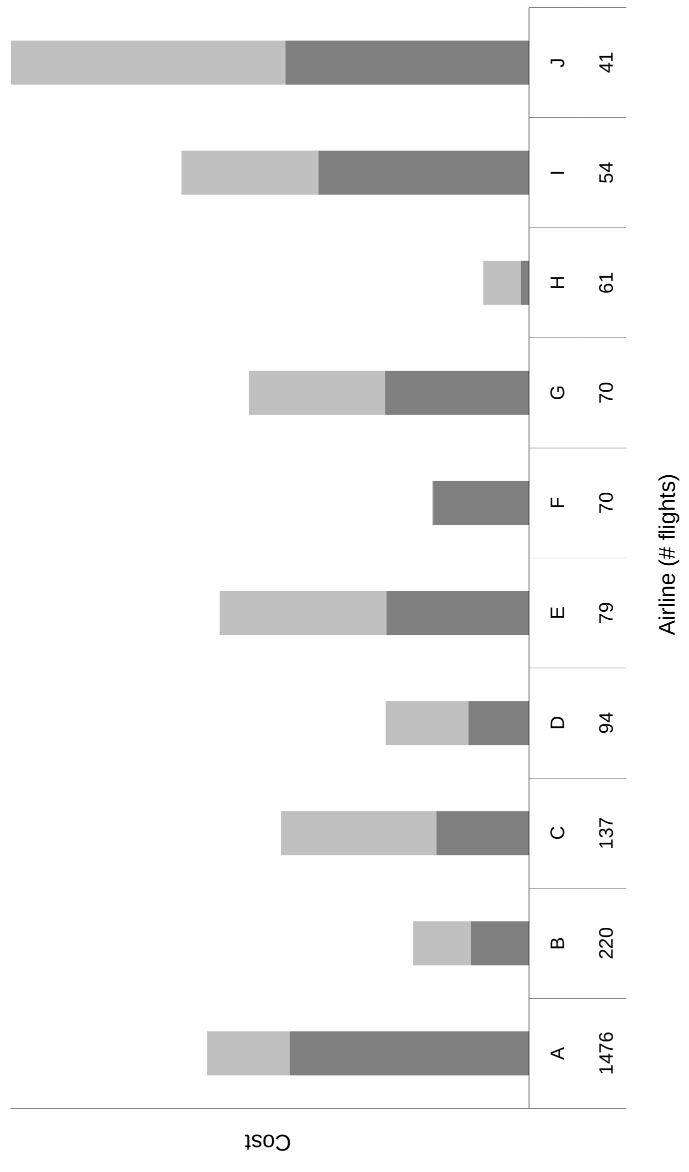


Figure 5.6: Average cost per flight in FCFS schedule (total column) and the local search schedule (dark gray)

In Figure 5.7 the total scaled cost resulting from the schedules generated by the eight heuristics is shown by the black bars. The heuristics obtain savings from 19% to 42% in terms of scaled cost compared to the FCFS schedule. The heuristics focused on fairness ((3) - (8)) obtain less cost savings than heuristic (2).

In order to evaluate the absolute fairness the root mean square deviation of the average airline scaled cost $\sigma_{\bar{c}}$ as defined by equation (4.4) is depicted by the gray bars.

It is surprising that heuristic (5) does only marginally lower this airline root mean square deviation compared to heuristic (2). Since also the total scaled cost savings are not competitive, the conclusion is that this heuristic often gives sub optimal solutions. Heuristic (3), as expected, lowers the root mean square deviation more substantially, but is still dominated by heuristic (4), where both the total scaled cost and root mean square deviation are lower. This shows that aiming at relative fairness also improves absolute fairness. The results of heuristic (6) are very similar to the minimum cost schedule resulting from heuristic (2) that is used as initial solution. It seems not possible to substantially improve fairness starting from a low cost solution. Heuristic (7) uses scaled cost as secondary objective and it is therefore not surprising it obtains a relatively low total scaled cost. Heuristic (8) has the largest root mean square deviation, which shows that aiming for a fair distribution of delays not necessarily leads to a fair distribution of scaled cost.

In Figure 5.8 the total delay resulting from the schedules generated by the eight heuristics is shown by the black bars. The reductions in delay are small. This is expected, since the FCFS schedule inherently will not have large delays. The reduction of delay can only be obtained by sequences reducing the total separation time required. However, this can only be done by increasing the delay of at least one flight. As expected, heuristic (7) and (8) obtain the largest delay reduction (9%).

The gray bars depict the root mean square deviation of the average airline delay, $\sigma_{\bar{d}}$ as defined by equation (4.10). This measures the fairness measured by delay. Heuristic (3), (4), (7) and (8) reduce this root mean square deviation substantially.

In Figure 5.9 the total real cost resulting from the schedules generated by the eight heuristics is shown by the black bars. The gray bars depict the root mean square deviation of these cost. The results are similar to that of the scaled cost. There is only a large difference for heuristic (8). This follows

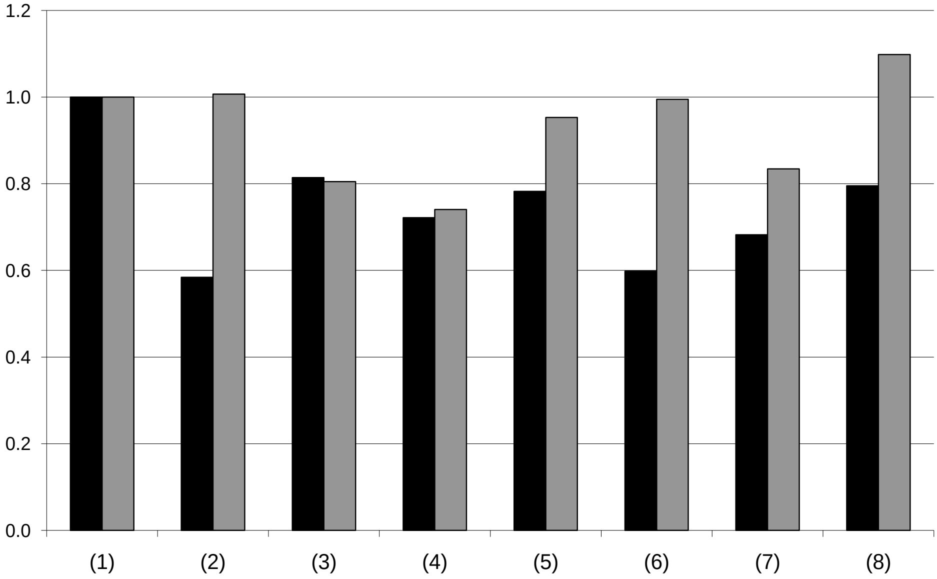


Figure 5.7: Total scaled cost and the root mean square deviation of the average airline scaled cost

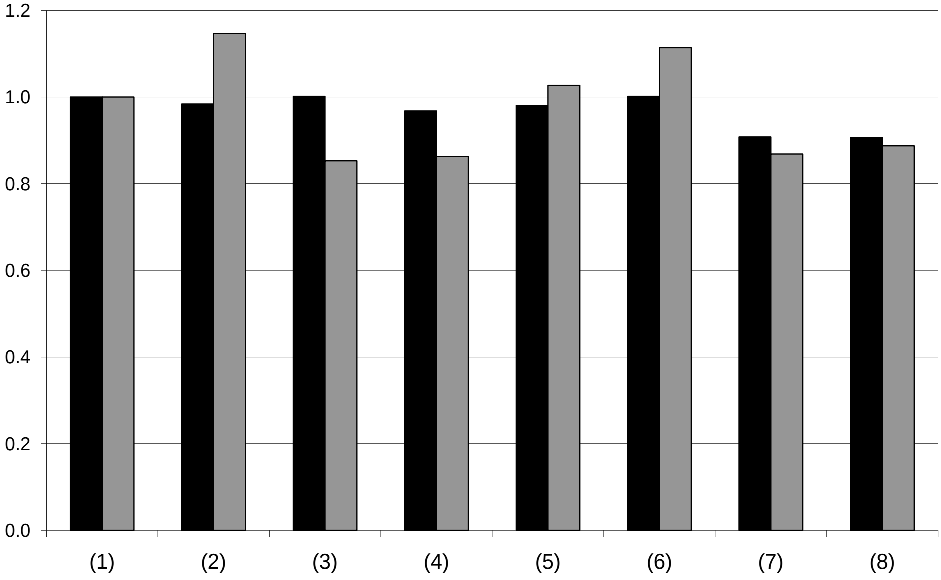


Figure 5.8: Total delay and the root mean square deviation of the average airline delay

from the explicit focus of this heuristic on the real cost (as trade-off to the fairness measured by delay). This heuristic gives the largest cost savings (38%) compared to the FCFS schedule. This is more than the minimum total scaled cost schedule which obtains 35% real cost savings.

In Figure (5.10) the percentage of airlines that have larger (real) cost compared to the FCFS schedule is shown, as defined in equation (4.7). This is a measure for the relative fairness. Heuristic (4) has the lowest percentage of airlines without any improvement (5%), as expected. Although heuristic (8) reduces the spread in the average delays of the airlines and has the lowest total real cost, it has the highest percentage of airlines that have larger (real) cost compared to the FCFS schedule (34%). So the fairer spread of delay does not necessarily lead to cost savings for all airlines.

Heuristic (6) indeed seem to improve the absolute fairness of its initial solution (the minimum total cost schedule resulting from heuristic (2)) slightly. It is still obtaining 40% total cost savings. This heuristic is suitable if total cost is the most important consideration and fairness only a secondary issue. It is dominated by the other heuristics on most fairness measures.

Heuristic (7) and (8) provide a large delay reduction and as expected a good fairness measured by delay. Surprisingly these heuristics provides also considerable cost savings. Heuristic (8) even obtains the lowest total real cost. However, these heuristics score the worst on relative fairness.

Heuristic (4) performs well on absolute fairness, relative fairness and fairness measured by delay. It is better than heuristic (1), (2), (5) and (6) on all these fairness measures. At the same time it obtains 30% savings in total scaled cost compared to the FCFS schedule. It can be concluded that this heuristic gives a good balance between the various criteria and may therefore be preferred over the others.

All these results were fairly similar for the different instances.

Results by airline category

To look into more detail to the fairness of the results, we defined 4 categories of airlines, based on the number of flights of the airline in the dataset. The first category consists of a single airline, the hub airline. This airline has almost half of the total number of flights. The large airlines have at least 70 flights per week (on average at least 10 per day). The medium airlines have at least 21 flights per week (on average at least 3 per day).

For each airline a the average cost per flight $\bar{c}_a(t_1, \ldots, t_N)$ can be cal-

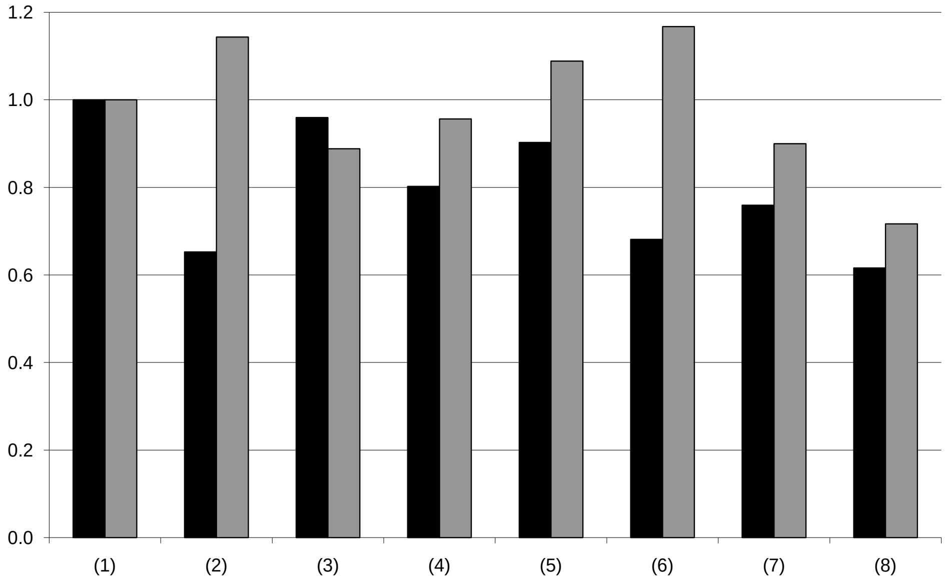


Figure 5.9: Total real cost and the root mean square deviation of the average airline real cost

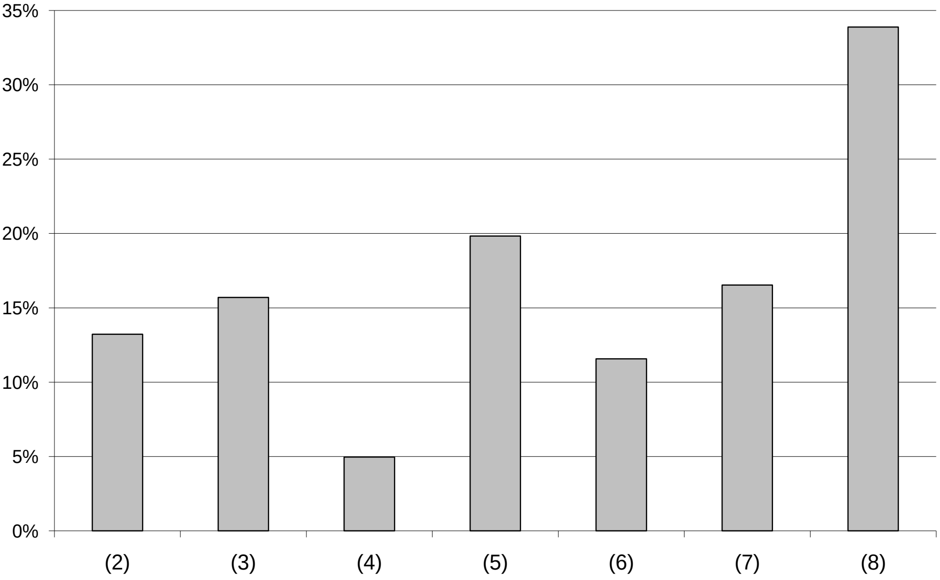


Figure 5.10: Percent of airlines that have no real cost improvement compared to the FCFS schedule

Category	Total Flights	Number of Airlines
Hub Airline	1880	1
Large Airlines	941	7
Medium Airlines	539	15
Small Airlines	618	98

Table 5.2: Airline categories

culated for a given schedule. These numbers are used to calculate airline average scaled cost for each category. Let A^m be the set of airlines in category m. The category average $\tilde{c}_m$ is now calculated by

$$\tilde{c}_m := \frac{1}{|A^m|} \sum_{a \in A^m} \bar{c}_a(t_1, \dots, t_N).$$

In Table 5.3 the normalized average category scaled cost are shown. For category m these are obtained by

$$\frac{\tilde{c}_m}{\frac{1}{4}\sum_{r=1}^{4} \tilde{c}_r}.$$

Heuristic	Hub airline	Large airlines	Medium airlines	Small airlines
(1)	0.68	0.95	1.19	1.19
(2)	0.79	0.85	1.19	1.16
(3)	1.07	1.02	0.87	1.04
(4)	0.85	1.07	1.03	1.04
(5)	0.92	0.90	1.05	1.12
(6)	0.85	0.82	1.14	1.19
(7)	0.80	0.88	1.14	1.17
(8)	0.38	1.04	1.24	1.33

Table 5.3: Normalized average scaled cost per airline category

It is clear that using FCFS schedule the medium and small airlines have larger scaled cost per flight on average. The costs are 16% and 19% larger than the average over the categories. This is because these airlines have a relatively large share of their flights during peak periods, in which delays are much more likely to occur.

In Table 5.4 the real cost for the airlines in each category divided by the real cost in the category using the FCFS schedule as obtained by heuristic (1) are shown. For category m this is calculated by

$$\frac{\sum_{a\in A^m}\sum_{i\in F_a}\kappa_i(t_i)}{\sum_{a\in A^m}\sum_{i\in F_a}\kappa_i(\hat{t}_i)}.$$

Heuristic	Hub airline	Large airlines	Medium airlines	Small airlines
(1)	1.00	1.00	1.00	1.00
(2)	0.67	0.53	0.65	0.83
(3)	1.13	0.78	0.54	0.72
(4)	0.85	0.77	0.64	0.76
(5)	1.00	0.69	0.72	0.87
(6)	0.73	0.50	0.65	0.82
(7)	0.80	0.64	0.68	0.77
(8)	0.52	0.90	0.83	0.72

Table 5.4: Relative real cost compared to heuristic (1) (FCFS schedule)

Heuristic (2) focuses only on total scaled cost minimization. The average improvement (in real cost) per category lies between 17% for the small airlines and 47% for the large airlines.

Heuristic (3) especially reduces the cost 46% for the medium and 28% for the small airlines. This however comes with a cost increase for the hub airline. Overall it results in the category average scaled cost being much closer together.

Using heuristic 4, the average percentage improvements for the categories are closer together. The improvement is between 15% for the hub airline and 36% for the medium airlines.

Heuristic (8) focuses on real cost instead of scaled cost. The different cost structure of the hub airline has a large impact on the results. The real cost of delaying a flight (with a considerable number of transfer passengers) of the hub airline are higher than of other airlines, because of the cost related to the missed transfers. Without scaling this leads to larger absolute and relative cost savings for the hub airline. However, the average real cost per flight are still (a little) above average for the hub airline. The average delay is nearly equal over the categories.

The airlines that are worse off than in the FCFS schedule fall mostly in the category of small airlines. This is directly related to their small number of flights. Twenty airlines in this category have only one flight in the dataset. Some of these flights are scheduled on time in the FCFS schedule and have a (small) delay in some of the schedules obtained using the other heuristics. The percentage of airlines that are worse off in this category flights is between 6% with heuristic (4) and 22% with heuristic (7). Since this still represents a minority of the airlines in this category, we expect that all airlines would achieve cost savings over longer periods of times (where these airlines will have a larger total number of flights). That is due to the fact that the operational circumstances will be different from week to week and it is unlikely that the same flights will always receive delays.

5.3.4 Dynamic Scheduling

Our formulation is deterministic, which means that all parameter values are assumed to be known before running the algorithm. In practice, unexpected departure delays and weather changes occur during operations. Consequently, possible landing intervals and required separation times may change. Our formulation and heuristic allow for a dynamic use of the model by recalculating the schedule, every time the circumstances change. It is interesting to evaluate whether such a dynamic use still yields cost savings.

In addition to the costs incurred by the schedule accomplished in practice, it is also important to consider the robustness of the schedules with respect to rescheduling. An updated schedule has to be communicated to those pilots that have to adjust their speed in order to arrive at a different time. Although pilots have assured us that such speed changes are unproblematic, it is preferable to have only few changes during the flights.

In this section the use of heuristic (2) in a dynamic setting is evaluated using a simulation. In a simulation run, a randomly generated timetable for a period of two hours is used. We tested timetables with different characteristics. The expected number of arrivals per hour was 20, 30 or 35 and the percentage of intercontinental flights was 10, 30, 50 or 70%. The latter is included because different flight lengths might have different effects: A (small) departure delay of an intercontinental flight can be absorbed en-route, while this may not be possible for a local flight. The latter can cause the latest schedule, which was obtained before the delay was known, to be infeasible.

Combining the different values for these parameters gives 12 different settings. For each of these settings a simulation with 100 runs was performed.

The timetable inter-arrival times are drawn from an exponential distribution, with a mean corresponding to the expected number of arrivals considered. A flight is intercontinental with a probability corresponding to the percentage of intercontinental flights in the current setting. An intercontinental flight has a uniformly distributed flight length between 5 and 15 hours and the local flights between 1 and 4 hours. The weight category of the flights is randomly distributed with 35 % heavy, 60 % medium and 5 % small aircraft.

It is assumed that there are 4 airlines. A flight belongs to airline A,B,C or D with probability 0.5, 0.25, 0.15 or 0.1, respectively. Airline A uses the airport as a hub and its costs are determined by the number of missed transfers. The cost functions for flights of this airline were generated in the same way as in the static experiment. The main consideration of the other airlines is punctuality and their cost functions are also determined in the same manner as in the static experiment.

All simulation runs start before the earliest departure in the generated timetable. The local search algorithm is used to create an initial arrival schedule for the flights. It is assumed all flights have a maximum departure delay of 3 hours.

At the scheduled departure time of a flight, a departure delay (or earliness) for the flight is determined randomly. This delay is uniformly distributed between 5 minutes early and 10 minutes late with probability 0.85 and larger than 10 minutes (on average 15 minutes), exponentially distributed, with probability 0.15.

The renewed arrival time interval is now determined by this departure time, the speed range and fuel reserve of the aircraft. The speed range of an aircraft is approximately between 5% below and 5% above its normal cruise speed. A flight has usually an extra fuel reserve of about 3%. There is also an emergency reserve, which however, cannot be considered for planning purposes. The fuel consumption depends on the speed of the aircraft. The normal cruise speed is the most fuel efficient.

Every 30 minutes the landing time intervals are updated, using the remaining distances, speed ranges and fuel reserves of the flights. If the last determined arrival schedule is not feasible or optimal with respect to the updated arrival time intervals, a new schedule is determined, using the local search algorithm. A simulation run ends when the simulation clock has reached the time all flights are landed.

In figures 5.11, 5.12 and 5.13 the average arrival delay (difference between timetable and actual arrival time) and scaled cost per flight are shown for

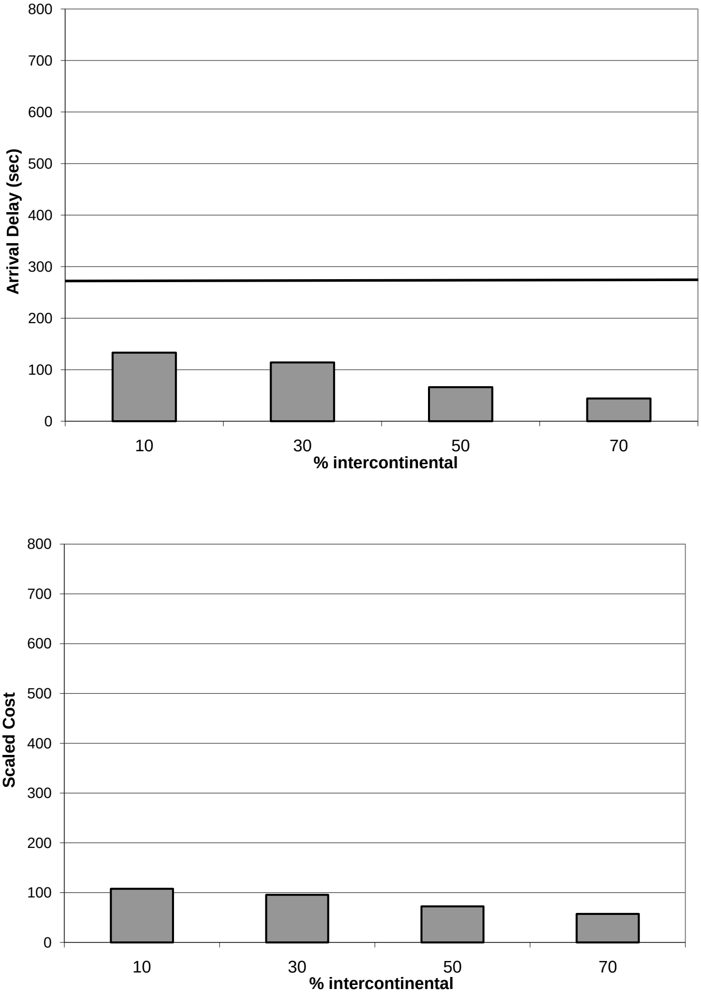


Figure 5.11: Mean arrival delay and scaled cost with 20 arrivals per hour

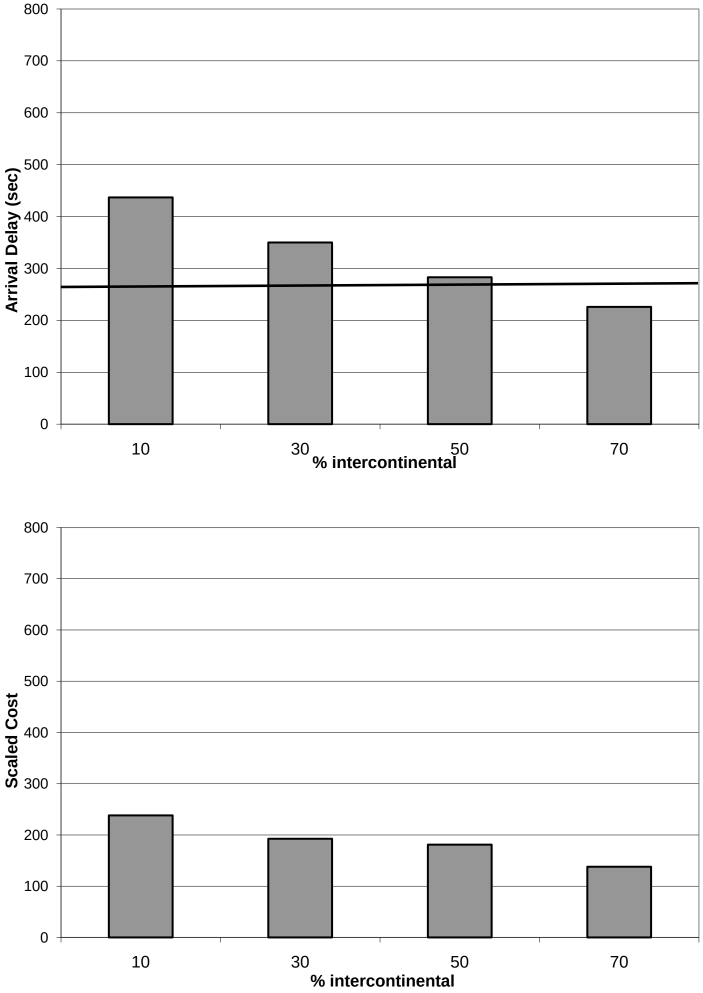


Figure 5.12: Mean arrival delay and scaled cost with 30 arrivals per hour

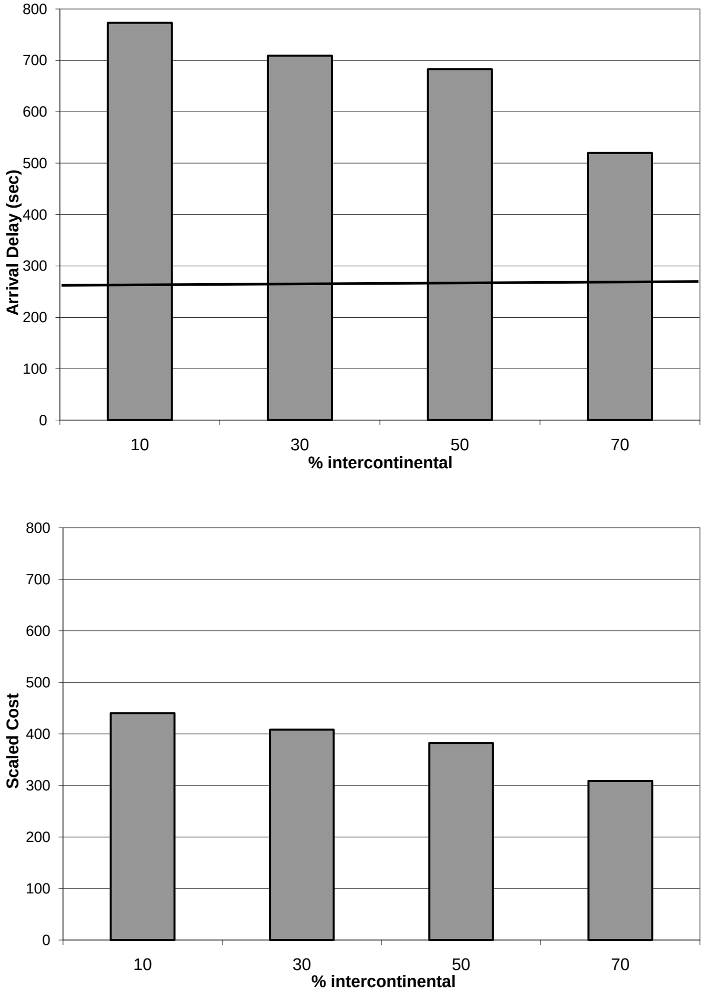


Figure 5.13: Mean arrival delay and scaled cost with 35 arrivals per hour

the different settings.

The average departure delay is just over 4 minutes (262.5 seconds). When there are only 20 arrivals per hour, most of this departure delay is consumed en-route and the average arrival delay is much smaller. When coming close to peak capacity (30 or 35 arrivals per hour) the disruptions at the departure propagate to other flights. Departure delays of intercontinental flights can be (partly) absorbed en-route more frequently. This is reflected in the lower arrival delays when the percentage of intercontinental flights is larger. As expected, increasing arrival rates lead to increasing congestion at the runway, as reflected in the arrival delays depicted in figures 5.11, 5.12 and 5.13. However, these figures also show a relatively much lower increase in scaled cost. This shows that the dynamic use of the heuristic is still very cost effective, since all cost functions are convex.

To evaluate the amount of change in the schedule during a flight, the following measure is used. Let t_i^k denote the arrival time of flight i in the k-th schedule of the run. Then the amount of change for this flight Δ_i is defined as follows:

$$\Delta_i = \sum_k |t_i^k - t_i^{k-1}|.$$

The average per flight for this measure is just over 2 minutes for 20 arrivals per hour, almost 5 minutes for 30 arrivals per hour and around 6 minutes for 35 arrivals per hour. These values are below the average arrival delays and around the average departure delay. This shows the schedules are quite robust and not much rescheduling is needed, except for the unavoidable handling of departure delays. This indicates that there is no need to include measures in the algorithm that limit the amount of changes explicitly, like in Beasley et al. [18]. These types of measures will affect the quality of the solutions.

5.4 Conclusions

In this chapter a mathematical programming formulation for the aircraft landing problem using airline costs was presented. Different objectives were introduced, focused on minimum total scaled costs, absolute fairness, relative fairness and fairness measured by delay. Local search heuristics to obtain reasonable solutions for these formulations using short computation times were introduced.

A large number of instances, created using schedule data from a major European hub, were tested in computational experiments. The results show tremendous cost savings for the airlines compared to a schedule that resembles current practice, especially under low visibility conditions. This shows that the negative impact of delays on airlines and their passengers can indeed be reduced by considering airline preferences.

By using different (cost and fairness) objectives schedules with different trade-offs between efficiency, cost, delay, absolute and relative fairness can be obtained. The heuristic using the objective that focuses on relative fairness also performs well on several other criteria considered, such as total cost savings and absolute fairness. Although this heuristic does not dominate all other heuristics on all criteria, it provides a good balance between the criteria. Therefore it might be preferred over the other heuristics. Which trade-off is preferred eventually is a subject for further discussion among air traffic stakeholders.

A simulation experiment shows that our method can also be used dynamically in a practical setting. In such a setting the effects of (external) operational disruptions are incorporated by rescheduling regularly. These disruptions can be divided over the flights in a cost effective manner. The regular rescheduling leads to only minor changes in the landing times. This indicates that our method generates quite robust schedules.

Chapter 6

Using Airline Cost in Hub Airport Runway Operations Scheduling

6.1 Introduction

Many (large) airlines use a so-called hub and spoke network to make efficient use of their resources (aircraft and crew). This entails consolidating traffic from a diverse range of origins to a diverse range of final destinations through large hub airports. Hub airlines schedule subsequent series of arrivals and departures to maximize customer choice (transfer possibilities) and minimize customer travel times. These series are called banks. This leads to higher load factors and thus increases revenues. On large hubs it is not uncommon to have more than five of these banks on a day. During these periods, the demand of the airport is often close to or even exceeds the capacity.

In the previous chapter, a model was presented to schedule landings on a single runway (the aircraft landing problem) considering airline cost related to flight delays. In this chapter the model is extended to schedule landings and take-offs at multiple runways. This gives the possibility to explicitly consider hub airline operations. Again, this is done using the approach to consider airline cost in a fair manner that was introduced in Chapter 4. In Section 6.2.1 an additional type of cost function is introduced to represent airline cost related to dependencies between arriving and departing flights, such as cost related to missed transfers. The cost function scaling mechanism is extended to handle these cost functions.

In Section 6.2.2 a MIP formulation of the model is given. This formulation allows us to optimize the runway assignment of the flights. A local search heuristic to efficiently obtain reasonable schedules is introduced in

Section 6.3. For large instances, a rolling horizon approach can be used together with the local search heuristic. A problem-specific rolling horizon approach is introduced in Section 6.4.

In Section 6.5 the computational experiments are discussed. A simulation experiment to identify the impact of runway assignment optimization is discussed in Section 6.5.1.

Computational experiments using data from a large European hub airport were performed and are discussed in Section 6.5.2. The impact of scheduling arrivals and departures simultaneously during peak periods is evaluated using these experiments.

This chapter ends in Section 6.6 with a number of conclusions.

This research was originally presented in Soomer [69].

6.2 Model

The model is used to determine runway (landing and take-off) sequences and landing and take-off times for the flights at the runways of an airport. The landing and take-off times are constrained to be within predefined time windows and to allow for the required separation between the flights.

It is assumed that each flight has a predefined time window in which it has to land or take off at the hub. For arriving flights this can be determined using the remaining flight time and the amount of fuel left or a maximum departure delay at its origin. For departing flights it depends on the scheduled departure time and possible delays (e.g., the aircraft cannot be ready in time) and a maximum acceptable (additional) delay.

It is assumed that the runways (that are used simultaneously) are independent. This means that the flight paths of the runways are separated sufficiently, such that the runways can be treated independently. Consequently, the separation rules have to be applied only between flights using the same runway. Such a runway configuration is used at a lot of large (European) hub airports.

The model formulation considers both segregated and mixed runway use. In the latter landings and take-offs are combined at the same runway. In segregated mode, landing and take-off operations are not combined on the same runway at the same time.

When multiple runways are used, it has to be decided which flights uses which runway. In practice, the runway used by a flight is usually determined by its route and the location of the runways. However, at some airports it is not trivial to determine which runway will be used for each flight. In this case, runway assignment must be done. This can be incorporated in the optimization process or can be done beforehand.

Next, the representation of airline costs and the scaling mechanism is introduced. In Section 6.2.2 a MIP-formulation of the model is presented.

6.2.1 Airline Cost and Fairness

In Section 4.1 flight cost functions were introduced that represented cost (directly) related to flight delay. Since each flight has different characteristics, the airline is allowed to provide a different cost function for each individual flight. Such a function relates the runway operation time to cost. We will call these the single flight cost functions. The (delay) cost for scheduling flight i at time t are represented by its single flight cost function $\kappa_i(t)$.

In this section, a second type of cost function will be introduced. These cost functions represent cost related to the dependence between an arriving and departing flight (of the same airline). These costs will depend on the actual connection time between the two flights. The actual connection time determines, for example, if transfers passengers are able to make the connection. The flights could also use the same crew or aircraft. Too little or too much time between these flights can be inefficient or even make the current aircraft or crew assignments infeasible.

The airline is also allowed to provide a cost function κ_{ij} for each pair of an arriving flight i and a departing flight j. Such a function relates the time difference between the arrival and departure (take-off time minus landing time) to costs. We call these the flight pair cost functions. The costs related to the connection time when scheduling an arriving flight i at time t_i and a departing flight j at time t_j are represented by their flight pair cost function $\kappa_{ij}(t_j - t_i)$.

Again, fairness has to be considered and therefore the same restrictions on the shape of the cost functions are used: The functions are required to be convex and piecewise linear and to have a minimal cost of zero at a time within the interval they are defined on.

We will also apply the scaling mechanism, introduced in Section 4.1. However, this scaling mechanism must be adapted to include the flight pair cost functions.

The scaling mechanism ensures that the average cost per time unit per flight are approximately the same for all airlines. This is done by introducing a single scaling factor for each airline. All cost functions for flights from this airline are multiplied with this scaling factor. In this way the original cost ratio between flights of the airline is preserved.

Let us make this more precise. The following notation will be used:

Let $F = \{1, \dots, N\}$ be the set of all flights to consider.
Let A be the set of all airlines.
Let $F_a \subset F$ be the set of flights of airline $a \in A$. Note that $F = \bigcup_{a \in A} F_a$ and $F_a \cap F_b = \emptyset$, for all $a, b \in A, a \neq b$.
Let F_a^{ARR} be the set of all arriving flights of airline $a \in A$.
Let F_a^{DEP} be the set of all departing flights of airline $a \in A$. Note that $F_a = F_a^{ARR} \cup F_a^{DEP}$.
Let $\kappa_i(t)$ be the cost function, relating the runway operation time with cost, for flight $i \in F$.
Let $\kappa_{ij}(\tau)$ be the cost function, relating the connection time τ between an arriving flight $i \in F_a^{ARR}$ and a departing flight $i \in F_a^{DEP}$, $a \in A$.
Let $f_i(t)$ be the scaled cost function, relating the runway operation time with scaled cost, for flight $i \in F$.
Let $f_{ij}(t)$ the scaled cost function, relating the connection time τ between an arriving flight $i \in F_a^{ARR}$ and a departing flight $j \in F_a^{DEP}$, $a \in A$.
Let E_i and L_i be the earliest and latest possible runway operation times of flight $i \in F$, respectively.

For each airline $a \in A$ a scaling factor α_a is calculated. The single flight cost function $\kappa_i(t)$ will be scaled to the function

$$f_i(t) := \alpha_a \kappa_i(t) \qquad i \in F_a, a \in A.$$

A flight pair cost function $\kappa_{ij}(\tau)$ will be scaled to the function

$$f_{ij}(\tau) := \alpha_a \kappa_{ij}(\tau) \qquad i \in F_a^{ARR}, j \in F_a^{DEP}, a \in A.$$

The (scaled) single cost function $\kappa_i(t)$ ($f_i(t)$) for flight i is defined on the interval $[E_i, L_i]$. This means that that the (scaled) flight pair cost function $\kappa_{ij}(\tau)$ ($f_{ij}(\tau)$) for an arrival i and a departure j, where τ is the time difference between the flights ($t_j - t_i$), is defined for τ in the interval $[E_j - L_i, L_j - E_i]$.

The scaling factors α_a, $a \in A$ are determined such that:

$$\frac{1}{|F_a|}\left(\sum_{i\in F_a}\frac{\int_{E_i}^{L_i}\alpha_a\kappa_i(t)dt}{(L_i-E_i)^p}+\sum_{i\in F_a^{ARR}}\sum_{j\in F_a^{DEP}}\frac{\int_{E_j-L_i}^{L_j-E_i}\alpha_a\kappa_{ij}(\tau)d\tau}{(L_j-E_j+L_i-E_i)^p}\right)=1.$$

So,

$$\alpha_a=|F_a|\left(\sum_{i\in F_a}\frac{\int_{E_i}^{L_i}\kappa_i(t)dt}{(L_i-E_i)^p}+\sum_{i\in F_a^{ARR}}\sum_{j\in F_a^{DEP}}\frac{\int_{E_j-L_i}^{L_j-E_i}\kappa_{ij}(\tau)d\tau}{(L_j-E_j+L_i-E_i)^p}\right)^{-1},$$

where p is a parameter to minimize the effect of differences in the length of the runway operation time intervals. With $p = 1$ the scaling makes the average scaled cost per flight per time unit equal to 1 for all airlines. This gives an advantage to airlines that provide more scheduling flexibility by larger possible time intervals. It is preferable to choose p equal to 2 to correct for the difference in interval lengths or to choose p just over 2, to give a small flexibility reward for airlines with flights with relatively large average time intervals.

6.2.2 MIP Formulation

In this section a Mixed Integer Programming (MIP) formulation of the model is given. This formulation is an extension of the single runway formulation given in Section 5.1.

Let $F = \{1, \dots, N\}$ be the set of all flights to schedule. We can partition this set into the set of arriving flights F^{ARR} and departing flights F^{DEP}. These sets can be partitioned again according to their airline $a \in A$: F_a^{ARR} and F_a^{DEP} are the sets containing respectively all arrivals and departures from airline a.

Let

R : Number of available runways

E_i : Earliest possible runway operation time for flight i $\quad i \in F$

L_i : Latest possible runway operation time for flight i $\quad i \in F$

S_{ij} : Required separation time when flight i uses the same runway before flight j $\quad i, j \in F, i \neq j$

R_i : Set of runways which flight i can use $\quad i \in F$.

The main decision variables are the assigned runways and runway operation times. The formulation requires the following decision variables:

$$
\begin{array}{lll}
t_i & : \text{runway operation time for flight } i & i \in F \\
\delta_{ij} & = \begin{cases} 1 & \text{if flight } i \text{ is scheduled earlier than flight } j \\ 0 & \text{otherwise} \end{cases} & i, j \in F, i \neq j \\
\zeta_{ij} & = \begin{cases} 1 & \text{if flight } i \text{ and } j \text{ use the same runway} \\ 0 & \text{otherwise} \end{cases} & i, j \in F, i \neq j \\
\gamma_{ir} & = \begin{cases} 1 & \text{if flight } i \text{ uses runway } r \\ 0 & \text{otherwise} \end{cases} & i \in F, r = 1, \ldots, R
\end{array}
$$

To make sure the variables act as described, the following constraints are introduced:

$$E_i \leq t_i \leq L_i \qquad i \in F \tag{6.1}$$

$$\delta_{ij} + \delta_{ji} = 1 \qquad i, j \in F, j > i \tag{6.2}$$

$$\sum_{r \in R_i} \gamma_{ir} = 1 \qquad i \in F \tag{6.3}$$

$$\zeta_{ij} = \zeta_{ji} \qquad i, j \in F, j > i \tag{6.4}$$

$$\zeta_{ij} \geq \gamma_{ir} + \gamma_{jr} - 1 \qquad i, j \in F, j > i, R = 1, \ldots, R \tag{6.5}$$

Constraint (6.2) ensures that either flight i uses the runway before flight j or the reverse. These variables are needed in the separation constraints (which are introduced below).

Constraint (6.3) ensures that flight i is assigned to exactly one runway included in the set R_i.

Constraint (6.5) makes sure that if flight i and j both use runway r ($\gamma_{ir} = 1$ and $\gamma_{jr} = 1$) the variable ζ_{ij} must be 1.

Separation constraints between flights using the same runway are considered. To achieve this,the following sets of pair of flights, defined by their possible runway operation time intervals, are introduced:

U : the set of pairs (i, j) of flights for which it is uncertain whether flight i uses the runway before flight j

V : the set of pairs (i, j) of flights for which flight i definitely uses the runway before flight j, but for which the separation is not automatically satisfied

W : the set of pairs (i, j) of flights for which aircraft i definitely uses the runway before flight j, and the separation is automatically satisfied

More formally:

$$\begin{aligned}
U &= \{(i,j)| \quad E_j \leq E_i \leq L_j \text{ or } E_j \leq L_i \leq L_j \text{ or} \\
&\qquad\qquad\quad E_i \leq E_j \leq L_i \text{ or } E_i \leq L_j \leq L_i, i,j \in F, i \neq j,\} \\
V &= \{(i,j)| \quad L_i < E_j \text{ and } L_i + S_{ij} > E_j, i,j \in F, i \neq j\} \\
W &= \{(i,j)| \quad L_i < E_j \text{ and } L_i + S_{ij} \leq E_j, i,j \in F, i \neq j.\}
\end{aligned}$$

The following constraints will ensure the proper separation (if the flights use the same runway):

$$\delta_{ij} = 1 \qquad (i,j) \in V \cup W \tag{6.6}$$

$$t_j \geq t_i + S_{ij}\zeta_{ij} \qquad (i,j) \in V \tag{6.7}$$

$$t_j \geq t_i + S_{ij}\zeta_{ij} - (S_{ij} + L_i - E_j)\delta_{ji} \qquad (i,j) \in U \tag{6.8}$$

If flight i definitely precedes flight j then we can fix δ_{ij} (constraint (6.6)). For $(i,j) \in V$ the proper separation still needs to be ensured if both flights land on the same runway (constraint (6.7)). This must also be done for the pairs in U. This is done by constraint (6.8) for the pair (i,j) if flight i lands before flight j ($\delta_{ij} = 1, \delta_{ji} = 0$) on the same runway ($\zeta_{ij} = 1$). If this is not the case this constraint is superfluous. Note that if $(i,j) \in U$ then $(j,i) \in U$ and constraint (6.8) ensures the separation for both orders.

Our objective will be the sum of all scaled cost functions:

$$\sum_{i \in F} f_i(t_i) + \sum_{a \in A} \sum_{i \in F_a^{ARR}} \sum_{j \in F_a^{DEP}} f_{ij}(t_j - t_i)$$

All these cost functions are piecewise linear and thus not necessarily linear in the current decision variables t_i. Therefore, we introduce some additional decision variables:

c_i : direct scaled costs involved with scheduling flight i at time t_i (value of scaled single flight cost function), $i \in F$

c_{ij} : scaled costs involved with scheduling arrival i and departure j with a time difference $t_j - t_i$ (value of scaled flight pair cost function), $i \in F_a^{ARR}, j \in F_a^{DEP}, a \in A$

That is, c_i should be equal to $f_i(t_i)$ and c_{ij} to $f_{ij}(t_j - t_i)$. To accomplish this we use the following observation. Consider a convex piecewise linear function $f(x)$ with K breakpoints. This function can be written as a set of $(K+1)$ linear functions with slopes $A_0, \ldots, A_K$ and intercepts $B_0, \ldots, B_K$:

$$f(x) = \begin{cases} A_0 x + B_0 & 0 \leq x \leq X_1 \\ A_1 x + B_1 & X_1 \leq x \leq X_2 \\ \vdots & \vdots \\ A_K x + B_K & X_K \leq x \end{cases}$$

with $X_1, \ldots, X_K$ the breakpoints of the function. This function can be written as

$$f(x) = \max_{k=0,\ldots,K} \{A_k x + B_k\}. \tag{6.9}$$

A proof is given in Section 5.1.3.

Using the same notation (with additional indices) for the convex piecewise linear cost functions $f_i(t_i)$ and $f_{ij}(t_j - t_i)$ and equation (6.9), the following constraints are introduced to represent these cost functions:

$$c_i \geq A_{ik} t_i + B_{ik} \qquad i \in F, k = 0, \ldots, K_i \tag{6.10}$$

$$c_{ij} \geq A_{ijk}(t_j - t_i) + B_{ijk} \qquad i \in F_a^{ARR}, j \in F_a^{DEP}, \quad k = 0, \ldots, K_{ij}, a \in A \tag{6.11}$$

The objective is to minimize the total scaled cost:

$$z = \min \left[\sum_{i \in F} c_i + \sum_{a \in A} \sum_{i \in F_a^{ARR}} \sum_{j \in F_a^{DEP}} c_{ij} \right] \tag{6.12}$$

6.3 Local Search Heuristic

In Section 5.2 a local search heuristic for the single runway aircraft landing problem was introduced. This heuristic will be extended to include multiple runways and runway assignment.

First, the the case without runway assignment is considered. This means there is a predefined runway assignment, which will not be altered by the heuristic. For each runway, a sequence and the scheduled times (complying to the separation rules) of the flights should be determined. If the runway sequences are given, the MIP-formulation becomes an LP (because the values of all binary variables are known). This formulation consists of constraints (6.1), (6.7), (6.10) and (6.11) and the objective (6.12). The solution of this LP provides the optimal runway operation times given the runway assignment and sequences.

The idea behind our heuristic is to repeatedly find an improved sequence for one of the runways. This means a sequence for which the corresponding LP formulation has a lower optimal value than the previous formulation. To find such an improved sequence, local search is used.

The general local search algorithm is given below.

```
LOCAL SEARCH()
$S =$ initial feasible solution
while there is a neighbor of $S$ of better quality
    do $S =$ neighbor of $S$ of better quality
```

Next we will specify how to find an initial feasible solution, the definition of the neighborhood and the selection procedure for a neighbor of better quality. Some of these procedures are designed for a situation with segregated runway mode. This means that landing aircraft use a different set of runways than flights taking off. This mode is used at a many large (European) hubs, including the one that is considered in our experiments. This gives the possibility to schedule all arrivals during the arrival peak and the (connecting) departures during the subsequent departure peak simultaneously in the same instance, without considering the departures (runways) during the arrival peak and the arrivals (runways) during the departure peak. This limits the number of flights in an instance and decreasing computation times.

We can also define neighborhoods in which it is possible to alter the initial runway assignment. This is considered in Section 6.3.4.

6.3.1 Initial Feasible Solution

An initial sequence of the flights on each runway has to be chosen. We will introduce two types of initial sequences.

The first one is the FCFS sequence, which is discussed in detail in Section 5.2.1.

The second sequence is denoted the transfer sequence. This sequence can be used under the assumption of segregated runway use. First, the LP model for the FCFS sequence is solved. For the runways that are used for landings, the FCFS sequence is used as initial sequence. The *landing* times for the *arriving* flights in this sequence are obtained by solving the LP and will be used to determine an initial sequence for the departures.

If an arriving flight has a large delay and has a lot of transfer passengers, it might be cost effective to delay some of the departing flights it has (short) connections to. This possibility is considered in determining the initial departure sequences. Take-off times for departing flights are determined assuming that the arriving flights land at the FCFS landing times. The initial sequences for the take-off runways are determined by ordering the flights by these take-off times.

Let us make this more precise: Let t_i^* be the landing time for arriving flight i as obtained by solving the LP for the FCFS sequence. Let $\hat{t}_j$ be the departure time such that $f_j(\hat{t}_j) = 0$. Let $t_j^{(i)}$ be the departure time such that $f_{ij}(t_j^{(i)} - t_i^*) = 0$ for all arrivals i, where there is a flight pair cost function defined for flight i and j.

We will determine a minimum cost departure time t_j^* for all departures in the following way.

$$t_j^* = \arg\min_{t_j \in T_j} \left\{ f_j(t_j) + \sum_{i \in F_a^{ARR}} f_{ij}(t_j - t_i^*) \right\}$$

with

$$T_j = \{\hat{t}_j\} \cup \{t_j^{(i)} : i \in F_a^{ARR}\}$$

This means that the individual preferred take-off time and the take-off times corresponding to the ideal connection times with each related arrival (under the assumption of FCFS landing times) are considered as take-off times for the departure.

The initial sequence for the departures is determined by ordering the flights by increasing t_j^*.

Feasibility

By solving the LP it is checked if a feasible solution (runway operation times) given the initial sequence exists. If this is not the case the sequences have to be changed in order to find a feasible solution.

Infeasibility means that at one of the runways it is not possible to schedule some of the flights such that the scheduled time is before the latest possible runway operation time and the separation constraints are satisfied. We will try to create a feasible runway sequence by repeatedly swapping two adjacent flights in one of the infeasible sequences for which the earlier one has a larger latest runway operation time, until the LP of such a sequence is feasible.

If this still not gives a feasible solution, a new sequence is repeatedly obtained by swapping two adjacent flights, for which the total sequence require less separation.

A more detailed and formal description of this procedure can be found in Section 5.2.1.

6.3.2 Neighborhoods

A neighbor is defined as a solution where one of the runway sequences is altered in some predefined way. We can define different ways to choose which runway sequence will be altered and how this is done.

We could select the runways one at a time and do improvements until no improvement for the currently selected runway can be found and then proceed with the next runway. Another option is to select a (possibly different) runway in every iteration.

To alter a runway sequence we use the shift and swap operations that were discussed in Section 5.2.2. A short summary is given here.

A swap operation swaps the positions of two flights in the runway sequence. This can be done for pairs of flights with overlapping runway operation time intervals. The extension involves a possible change of position for flights positioned in between the swapped flights, which have a non-overlapping runway interval with one of the swapped flights.

Further we use a shift operation, which removes a flight and inserts it again at a different position. This can be done if the flight originally at the new position has an overlapping runway interval with the shifted flight. The extension involves a possible change of position for flights positioned in between the old and new position of the shifted flight, which have a non-overlapping runway interval with this flight.

6.3.3 Selection of a Neighbor

To reduce the computation time, it is preferable to select a neighbor in a manner that finds an improvement by evaluating as few neighbors as possible.

Therefore, we will evaluate promising neighbors first. To find promising neighbors the objective improvement of each neighbor is estimated. The calculation of the estimation should be very efficient, compared to an evaluation (solving the LP). The estimation uses the runway operation times in the current solution, to estimate the runway operation times and involved scaled cost after performing the operation. This can be done by estimating the cost for the flight at the i-th position in the neighbor by using the flights' cost function with the optimal time for the flight at the i-th position in the current solution.

First we have to choose which runway sequence(s) to consider. For the sequences considered, we have to estimate the improvement for all possible operations on the flights in the sequence(s). The neighbor with the largest estimated improvement (in all considered sequences) will be evaluated. If the LP-objective for this neighbor is smaller than the current objective, this neighbor is selected. Otherwise the neighbor with the second largest estimated improvement is evaluated, etc.

6.3.4 Runway Assignment

To enable runway assignment, the initial solution, neighborhoods and selection methods have to be adjusted somewhat.

There must be some initial runway assignment, which we can use to create an initial feasible solution. This initial assignment must be practically feasible and reasonably efficient. If this is the case, we can use the procedure from Section 6.3.1 to create an initial feasible solution, given the initial runway assignment. Such an initial assignment can for example be obtained by a round robin assignment of the flights to the runways.

The neighborhoods must include operations to change the assigned runway of some flight(s). The swap neighborhood can be extended with swaps between flights at different runways. Of course, both runways should be valid for both flights with respect to the type of operations. In case of a runway swap, both flights will basically be at the same position in the runway sequence as the other flight was before the swap. We will denominated this neighborhood the swap$^+$ neighborhood.

The shift neighborhood can be extended with shifts where a flight is removed from its original position and inserted at a position on a different runway. Of course, the new runway should be valid for the flight with respect to the type of operations. We will call this neighborhood the shift$^+$ neighborhood.

The neighbor selection procedure from the previous section can be used with these neighborhoods. In this case, an estimation for the improvement must be calculated for the neighbors that involve a runway change too.

6.3.5 Summary

Let π be the set of sequences for all runways. The local search algorithm can be summarized as follows:

```
Local Search()
π := Find initial solution
N(π) := Set of neighbors of π
Estimate objective improvement for all members of N(π)
while N(π) ≠ ∅
    do π' = neighbor with maximum estimated improvement in N(π)
        if LP(π') is feasible and z_{LP(π')} ≤ z_{LP(π)}
            then π = π'
                N(π) = Set of neighbors of π
                Estimate objective improvements for all members of N(π)
            else N(π) = N(π) \ π'
return π and the optimal runway operation times
```

6.4 Rolling Horizon Approach

When the number of flights to schedule becomes very large and the planning horizon lengthens, it is advisable to use a rolling horizon approach in combination with the local search heuristic. This situation occurs when large number of arrivals and connecting departures are scheduled integrally. A rolling horizon approach will reduce the computation times considerably. At the same time it fits the practical situation where later flights are scheduled in later iterations (possibly considering updated information). The rolling horizon approach is a new contribution compared to the solution approach presented in the previous chapter.

In general a rolling horizon approach divides the total planning horizon in smaller time windows. The problem for the earliest time window is solved first. Then the time window is advanced such that it still overlaps partly with the old time window. This means that a part of the flights will be

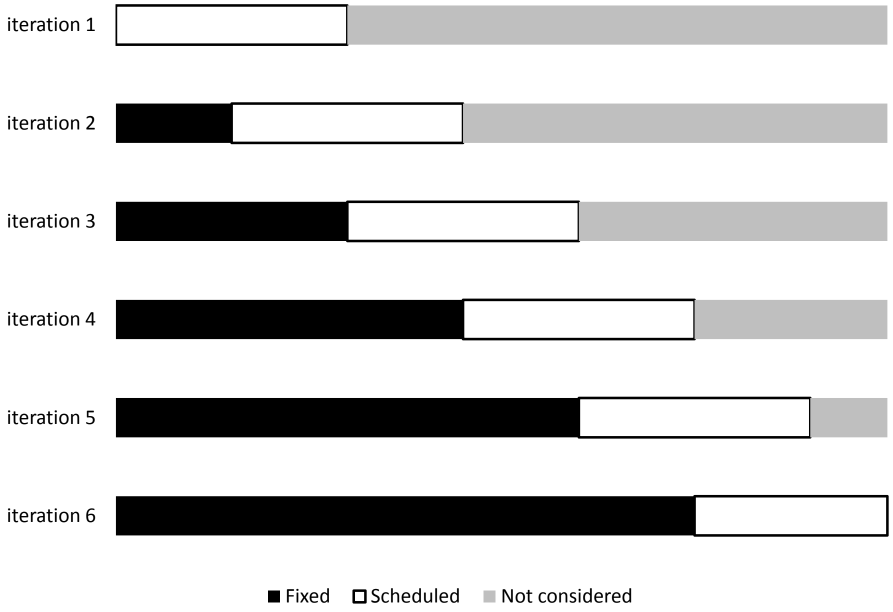


Figure 6.1: Rolling Horizon Approach

rescheduled together with the flights that are added by advancing the time window. This approach is depicted in Figure 6.1.

In the remainder of this section we will explain in detail how this rolling horizon approach is used to schedule an arrival peak and the subsequent departure peak. This approach assumes segregated runway use. When scheduling an arrival (that falls within the current scheduling horizon), the departures it has (short) connections to should be scheduled simultaneously. Thus, the trade-off between delays and missed transfers is considered. In order to achieve this, there will be different scheduling horizons for arrivals and departures.

For the arrivals, the horizon is advanced in the usual manner. The departure horizon is partly determined by including the departures that have a (short) connection with the arrivals in the current arrival horizon. In this way the connected flights are scheduled simultaneously in the same iteration and the connection costs are considered directly.

A connection between an arrival in the current arrival horizon and a departure can only be made if the departure is not scheduled earlier than the moment whereupon the minimum connection time has elapsed since the landing time of the arriving flight. An arrival included in the current arrival horizon could be scheduled at the starting time of the current arrival horizon (but not earlier). The earliest conceivable take-off time that allows for the connection to be made is therefore the moment whereupon the minimum connection time has elapsed since the starting time of the current arrival horizon. Consider the case in which the departure was scheduled at an earlier time in a previous iteration. The possibility to reschedule this departure in order to allow for the connection should be considered in the current iteration. At the same time, it could be the case that keeping this departure at its earlier time, incurs less cost to the airline (the cost of missing the connection are smaller than the delay costs involved with this departure). Therefore, the departure horizon should both include the original earlier departure time and the times that allow the connection to be made.

Let us make this more precise: First, an initial (fixed order) solution is calculated for the complete set of flights. Let t_i be the scheduled time in the current solution for flight i. Let T be the length of the considered arrival horizon. Let $[A, B]$ be the current planning horizon for arrivals and $[C, D]$ for departures. Initially, $A = C = D =: 0$ and $B := T$.

Let τ^*_{ij} be the time difference such that $f_{ij}(\tau^*_{ij}) = 0$ for an arrival i and a departure j that have a flight pair cost function.

The planning horizon for departures is updated by considering for each arrival i in the current arrival horizon($A \leq t_i < B$), each connected departure j. For every connection the departure horizon is updated in the following manner:

$$\begin{aligned} C &:= \min\{C, A + \tau^*_{ij}, t_j\} \\ D &:= \max\{D, B + \tau^*_{ij}\} \end{aligned}$$

After this, D is updated one more time, in the following way:

$$D := \max\{D, C + T\}$$

Now, all the arrivals i with $t_i \in [A, B]$ and all the departures j with $t_j \in [C, D]$ are scheduled simultaneously using the local search heuristic. After this, the arrival horizon moves forward by $\frac{1}{2}T$. So, $A := A + \frac{1}{2}T$, $B := B + \frac{1}{2}T$ and $C := C + \frac{1}{2}T$. The departure horizon will again be updated in order to consider the connections with the flights in the arrival horizon. The corresponding flights are added to the problem. For the arrivals i with a landing time $t_i < A$ the landing times are fixed. The same holds for the departures j with a take-off time $t_j < C$. Then, the local search heuristic is applied to the arrivals in the current arrival horizon and the departures in the current departure horizon simultaneously. This process will continue until all the flights are scheduled.

6.5 Computational Experiments

In this section, the computational experiments and its results are presented.

A simulation experiment was performed to assess the difference between fixed and dynamic runway assignment for landing flights. This experiment is discussed in Section 6.5.1.

Ten instances created using timetable data from a major European hub, were used to assess the results of scheduling (connecting) arrivals and departures integrally. This is discussed in Section 6.5.2.

6.5.1 Runway Assignment Simulation

These experiments were performed to assess the gains that can be obtained by optimizing the runway assignment. This is done by considering a set of arriving flights that will be scheduled. We assume that two runways are used for landings and all flights can land on each of the runways. In practice it often depends on the route and direction the flight is coming from on which set of the active runways a flight can land. However, in the experiments we want to evaluate the maximum gain that can be obtained by runway assignment.

A number of scenarios are defined, which differ in the average number of flights per hour and the separation rules used. The basic set of separation rules are the wake vortex standards from the International Civil Aviation Organization (ICAO) as listed in Table 6.1. Additionally a minimum separation required because of visibility conditions can be applied. In this case the maximum of the two separation distances should be used. The scenarios differ in the visibility conditions and the related separation required. A larger visibility separation will result in reduced runway capacity and therefore a higher load of the system.

For every scenario, 50 runs are performed. In every run flight data is generated randomly, using the scenario parameters. The data consist of the airline, timetable arrival time and cost functions for every flight. This data

		following aircraft		
		Light	Medium	Heavy
leading	Light	3	3	3
aircraft	Medium	5	3	3
	Heavy	6	5	4

Table 6.1: Wake vortex separation in nautical miles for different weight categories.

represents an hour of the timetable.

The timetable inter-arrival times are drawn from an exponential distribution, with a mean corresponding to the expected number of arrivals considered. The weight category of the flights will be randomly distributed with 35 % heavy, 60 % medium and 5 % small aircraft.

The single flight cost function of a flight has a minimum of zero at its timetable landing time. We will assume there are 4 airlines. A flight belongs to airline A,B,C or D with probability 0.5, 0.25, 0.15 or 0.1, respectively. Airline A uses the airport as a hub and its costs are mainly determined by the number of missed transfers. The main consideration of the other airlines is punctuality. The cost functions for flights of these airlines were generated in the same manner as in the experiments from Section 5.3.1.

In every run, schedules are obtained using different versions of the local search heuristic. The different schedules are the FCFS schedule and four schedules obtained by the local search heuristic using different neighborhoods. For the FCFS schedule, which is also used as initial solution in the local search, round robin runway assignment is used. This ensures that the load of both runways is approximately equal. The neighborhoods used to calculate the other schedules are the swap, swap$^+$, shift and shift$^+$ neighborhoods (see sections 6.3.2 and 6.3.4).

In Table 6.2 for each scenario, the sum of the scaled cost of the 50 runs as percentage of the sum of the FCFS scaled cost are shown. The neighborhoods with the "+" denote that runway changes were included in the neighborhoods. In the first two columns, the average number of flights per hour and the minimum visibility separation distance for the scenario are listed. A separation requirement of 8 nautical miles is not considered with 60 and 80 flights per hour, because this combination leads to a considerable overload of runway capacity.

The first conclusion is that the schedules calculated using the local search

Scenario		neighborhood			
flights	Separation	Shift	Shift$^+$	Swap	Swap$^+$
40	3	70%	64%	72%	65%
	4	70%	67%	72%	65%
	8	52%	50%	51%	47%
60	3	53%	51%	55%	49%
	4	42%	41%	41%	36%
80	3	31%	30%	33%	28%
	4	43%	41%	42%	39%

Table 6.2: Scaled cost as percentage of FCFS cost.

algorithm yield large savings in all scenarios, compared to the FCFS schedule. This reconfirms the results presented in Section 5.3.3.

The cost of FCFS schedules increases fast when the load of the system increases (either by number of flights or by larger separation requirements). The schedules obtained using the local search heuristics can partly absorb this increase, as can be seen from the increasing savings obtained.

As expected the possibility to change the runway assignment during the local search gives better results. However, the differences are smaller than expected, on average just over 2% of the FCFS cost using the shift neighborhood and just over 5% using the swap neighborhood. This indicates that the initial round robin runway assignment performs quite well. This is also confirmed in the paper of Bäuerle et al. [16].

6.5.2 Hub Airport Scheduling Experiments

In this section, experiments related to the combined scheduling of arrivals and departures are discussed.

Airport Data

Ten instances created using timetable data from a major European hub were tested. These instances represent peak periods on 6 different dates between May and October 2007. These peak periods consist of an arrival peak and a subsequent departure peak. This gives a large number of transfer possibilities for the passengers of the hub airline. In an instance we want to consider all the transfers between the arrivals in the arrival peak and the departures in the subsequent departure peak.

At the considered airport, runway assignment depends on the routes of a flight and is therefore not included in the optimization. Runways at this airport are not used in mixed mode. During an arrival peak usually two runways are used for arrivals and one for departures. This is reversed during a departure peak. The departures during the arrival peak and the arrivals during the departure peak do not necessarily have to be scheduled in the same instance because of the runway configuration. Therefore, the instances only consist of the arrivals during the arrival peak and the departures in the subsequent departure peak.

The ten instances contain 1775 flights in total, consisting of 871 arrivals and 904 departures. More than half of the flights (921) belong to the hub airline. Each instance contains between 135 and 210 flights. The following assumptions were made about the cost functions and possible arrival and departure times of the flights.

All arrivals can land between 15 minutes before the timetable arrival time and 150 minutes after the timetable arrival time. All departures can take off between 5 minutes before the timetable departure time and 180 minutes after the timetable arrival time. This might seem a little optimistic, but is done in order to be able to assess the potential savings that can be obtained by (starting) scheduling relatively long in advance.

All flights have a single flight cost function $\kappa_i(t)$ to represent the cost of delay. It is assumed that the single flight cost function has a minimum of zero cost at the timetable arrival or departure time of the flight. The average delay costs per minute are drawn from a uniform distribution between 1 and 9 for each flight. For every minute delay (compared to the timetable time) the costs increase with the generated delay cost per minute. For every minute the flight lands earlier than the timetable time, the costs increase with the generated delay cost divided by 10. An example of such a function is depicted in Figure 6.2.

The costs of the hub airline are for a large part determined by the number of missed transfers. The number of transfers between flights are not available in the data. Therefore, we will generate these in the following manner. For each arrival - departure pair of the hub airline the connection time is defined by the time between the timetable time of the arrival and the timetable time of the departure. There must be a minimum connection time of 45 minutes between the flights to allow for transfers. If the connection time is between 45 and 60 minutes, the average number of transfers between the two flights is

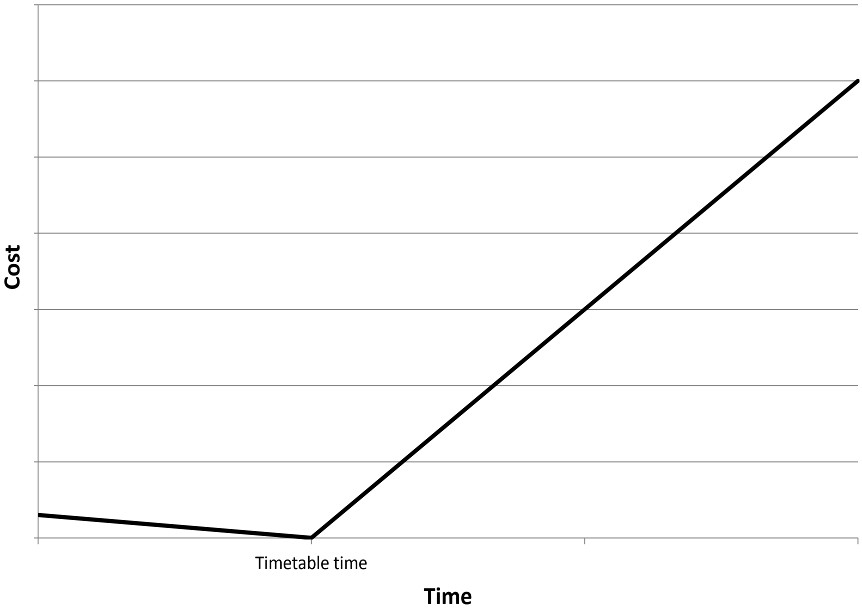


Figure 6.2: Single flight cost function used in the experiments.

3, decreasing to an average of 1 for flights with a connection time of 2 hours and more. The actual number of transfers are drawn from an exponential distribution with the above averages as parameter.

The slopes of the flight pair cost function of a pair of flights are determined by the generated number of transfers. If the actual time difference is smaller than the minimum connection time (45 minutes), the cost increases with the number of transfers with every minute less connection time. The costs will be zero in case of a 60 minute time difference, because this gives the passengers a convenient time period to transfer. Between 45 and 60 minute transfer time the cost will decrease with a slope equal to the number of transfers divided by 100. Similarly the cost will increase with with a slope equal to the number of transfers divided by 100 with a time difference between 60 and 120 minutes. With a longer connection time the slope is the number of transfers divided by 10, to represent the inconvenience of a longer waiting time. An example of such a function is depicted in Figure 6.3. Note that there is no direct dependency on the timetable connection time of the flights, because this has little impact on the actual experience of transfer

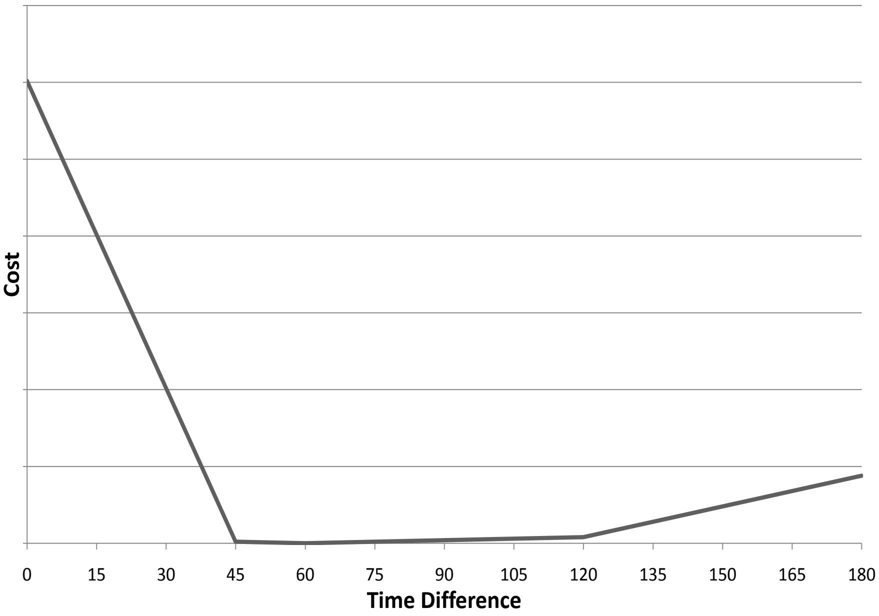


Figure 6.3: Flight pair cost function used in the experiments

passengers (especially in case of delays). In our instances, this timetable connection time is for most pairs less than 120 minutes and always between 45 and 150 minutes, which all give relatively low cost using these flight pair cost functions.

The hub airline uses both the single flight and flight pair cost functions as described above. The number of passengers connecting from or to a certain flight of the hub airline can range from 0 to over 100, where values between 40 and 80 are common. This number is on average (much) larger than the delay cost per minute. Since both values are used as slope for cost functions, this reflects the importance of missed transfers for the hub airline.

Applying the cost scaling mechanism gives the scaled cost function that will be used in the optimization process. The average scaled delay cost per minute per flight (as represented in the scaled single flight cost functions) of the hub airline will be (much) smaller than the average scaled delay cost per minute per flight of the other airlines. This is because the total airline costs (including the flight pair cost functions representing the missed transfer cost) are subject to the scaling mechanism. This represents the fact that the hub airline is less sensitive to delays that do not result in missed transfers.

Experiments

The experiments were performed under the assumption of bad visibility conditions, such that the minimum visibility separation distance is 6 nautical miles. In the peak periods, this results in a landing and take-off demand that exceeds the (reduced) landing and take-off capacity of the airport, resulting in delays. It is interesting to evaluate how these delays, the resulting costs and missed transfers are divided over the flights and airlines.

It is interesting to compare the difference in cost that results from the possibility to use flight pair cost functions. The flight pair cost functions are a natural way to model the hub airlines cost (w.r.t. missed transfers). Without the possibility to use of flight pair cost functions, the hub airline has to use another representation of these costs. This can be done by incorporating these costs in the single flight cost functions of the arrival.

It is very uncommon for a departure to take-off (much) earlier than its timetable time. This means that a connection is almost always made if the arriving flight lands before the timetable time of the departure minus the minimum connection time. This assumption is used to incorporate the connection cost in the single flight cost function of the arrival. This is done by increasing the slope of this function after this point in time with the number of transfers between the flights. However, because of the assumption about the departure time, the situation in which the departure is delayed is not represented realistically. In this case, the arrival could have a similar (additional) delay without destroying the connection.

Let $\tilde{\kappa}_i(t)$ denote the single flight cost functions adapted with the transfer cost. $\tilde{f}_i(t)$ represents the *scaled* single flight cost functions adapted with the transfer cost.

Consider, for example, an arriving flight i with single flight cost function $\tilde{\kappa}_i(t)$ and a departing flight j. Suppose there are λ passengers with a transfer from flight i to flight j. Let $\hat{t}_j$ be the timetable departure time of flight j and $\hat{\tau}_{ij}$ be the minimum connection time between flight i and j. Now, $\tilde{\kappa}_i(t) := \kappa_i(t) + \lambda(t - \hat{t}_j + \hat{\tau}_{ij})$ for $t > \hat{t}_j - \hat{\tau}_{ij}$.

Now, we can compare schedules obtained by using the adapted single flight cost functions and schedules obtained by using the flight pair cost functions together with the (original) single flight cost functions. Since the latter situation represents the hub airline cost more naturally, we will use this to measure the scaled cost of all schedules (afterward).

This means that we obtain a schedule by a heuristic that uses the adapted single flight cost functions as objective:

$$\min \sum_{i \in F} \tilde{f}_i(t_i)$$

Afterward, the obtained schedule will be evaluated using the flight pair cost and single flight cost functions. Let $\tilde{t}_i$ be the landing time obtained by the heuristic. Now we will evaluate the scaled cost by

$$\sum_{i \in F} f_i(\tilde{t}_i) + \sum_{a \in A} \sum_{i \in F_a^{ARR}} \sum_{j \in F_a^{DEP}} f_{ij}(\tilde{t}_j - \tilde{t}_i)$$

In this manner we will calculate a schedule that considers the flight in FCFS sequence and a schedule that is obtained by using the rolling horizon local search heuristic.

Two schedules are also obtained by using the flight pair cost functions in the objective:

$$\min \left\{ \sum_{i \in F} f_i(t_i) + \sum_{a \in A} \sum_{i \in F_a^{ARR}} \sum_{j \in F_a^{DEP}} f_{ij}(t_j - t_i) \right\}$$

Using this objective we will calculate a schedule that considers the flight in the transfer sequence and a schedule that is obtained by using the rolling horizon local search heuristic.

Those four schedules were computed for all ten instances. The shift neighborhood is used for the local search heuristics.

Results

In Figure 6.4 the total scaled cost relative to the FCFS schedule are shown for the four difference schedules. It is clear that using the local search heuristic with the flight pair cost functions gives the lowest cost. The cost savings are 40% compared to the FCFS schedule and 22% compared to the local search heuristic using the adapted single flight cost functions. This shows that the possibility to use flight pair cost functions indeed leads to larger cost savings.

The number of missed transfers is reduced drastically when the flight pair cost functions are used. This is shown in Figure 6.5. In total there are 27,133

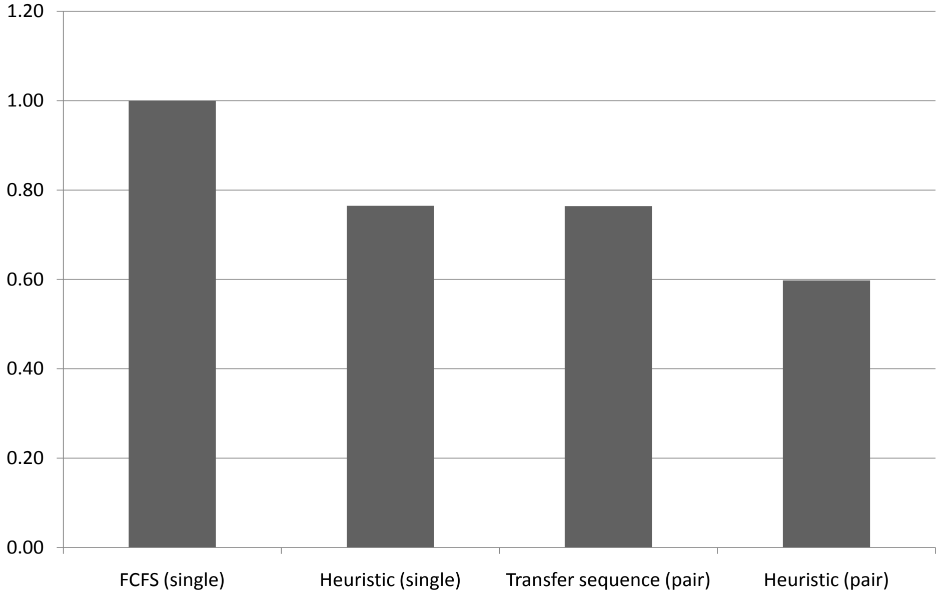


Figure 6.4: Total Scaled Cost using different heuristics

transfer passengers. A passenger misses its transfer if the connection time is less than 45 minutes. Using the FCFS schedule, 4947 of these passengers miss their connection. The local search heuristic using the adapted single flight cost functions reduces this to 2822. However, in the schedules that consider the flight pair cost functions the number of missed transfers are 996 and 1396.

The local search heuristic using the flight pair cost functions achieves 22% cost savings compared to the schedule using the transfer sequence.

The number of missed transfers is larger in the schedule obtained by the local search heuristic compared to the schedule using the transfer sequence. This may seem remarkable at first sight, but the latter sequence is explicitly designed to reduce the number of missed transfers. The local search heuristic is focused on minimizing the total scaled cost and indeed gives lower total scaled cost, at the expense of an increase in the number of missed transfers.

The hub airline has a different cost structure than the other airlines. Therefore it is interesting to look at the difference in average scaled cost per flight for the hub airline and the other airlines. This is shown in Figure 6.6.

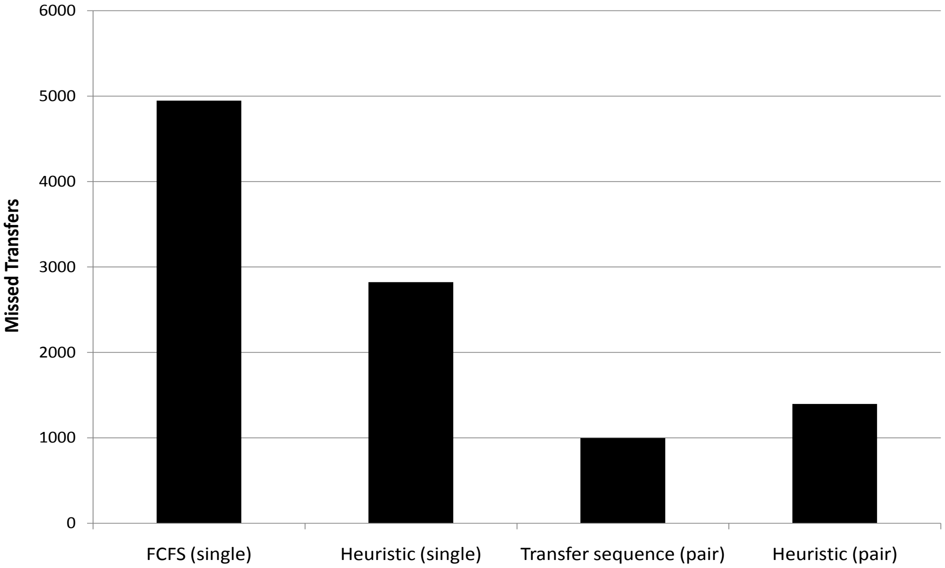


Figure 6.5: Total missed transfers using different heuristics

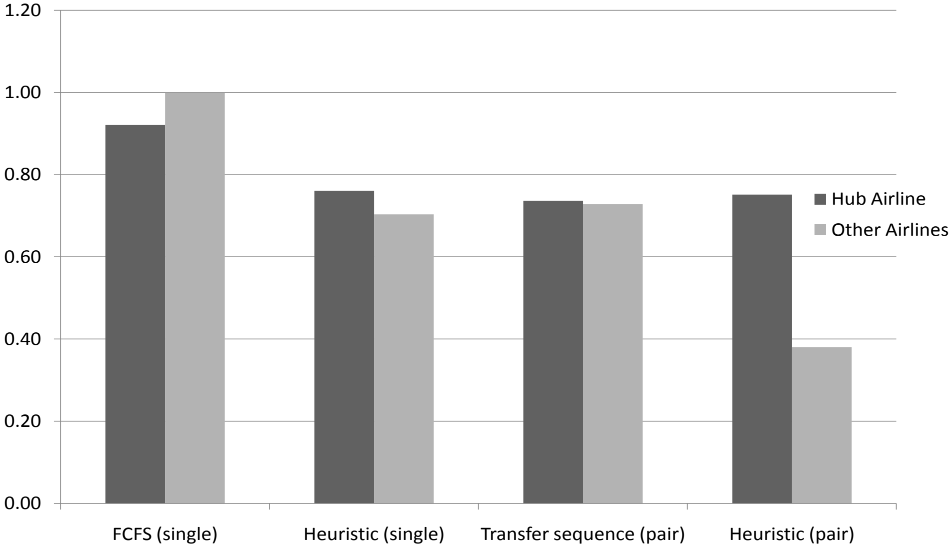


Figure 6.6: Average scaled cost per flight for the hub airline and other airlines using different heuristics

In the FCFS schedule there is only a small difference. This is explained by the almost equal average delay and the scaling mechanism.

It seems however that there is a limit to the amount of savings that can be obtained by the hub airline. This is explained by the large number of flights from this airline (and the reduced capacity scenario we consider) and the different cost structure of this airline. In this reduced capacity scenario, it is inevitable that a number of the arrivals of the hub airlines are delayed. Consider the impact these delays have on the connecting departing flights. These flights can either be delayed to wait for the transfer passengers or leave without the passengers. Both solutions lead to additional cost for the hub airline (either related to delay or missed transfers). It seems that in this situation the maximum savings of the hub airline are about 25% of the FCFS cost. This can be obtained either with less departure delays and more missed transfers using the single flight cost functions or with less missed transfers and more departure delay using the flight pair cost functions. Since transfer passengers are usually very important to the hub airline, the latter will be the most cost effective in most situations (depending on the exact number of transfer passengers and connection times).

All these results were fairly similar for the different instances.

6.6 Conclusions

In this chapter we discussed a model for hub airport runway scheduling (in peak periods). A peak period usually consists of an arrival bank followed by a departure bank. This provides many possibilities for transfers between flights of the hub airline. In a peak period the runway demand is usually close to or even exceeds runway capacity. This leads to potential delays. Both delays and missed connections have a large impact on airlines cost. In our model these airline cost are considered while scheduling runway operations. A framework for representing flight cost related to delays and cost related to connection times between arriving and departing flights using two types of cost functions is presented. Local search heuristics combined with a rolling horizon approach are used as a solution technique for realistically sized problems.

Simulation experiments were performed to assess the impact of dynamic runway assignment during the optimization process. The results show only a small improvement compared to a fixed round robin runway assignment.

Computational experiments using timetable data from peak periods at a larger European hub were performed to assess the impact of the simultaneously scheduling of arrival and (connecting) departures. The results show that using the cost representation of the interdependencies between these flights can cause considerable cost savings for the airlines. For the hub airline this could lead to a substantial decrease in the number of missed transfers.

Chapter 7

Summary and Conclusions

The purpose of the research discussed in this thesis was to assess the effects of considering airline preferences in runway operations scheduling. This was motivated by the congestion and delays that regularly occur in the air traffic system. Delays cause both cost and inconvenience to airlines and their passengers. Airborne delays increase fuel cost. Delays can cause infeasibility to crew and aircraft assignments for subsequent flights. In this way, delays are propagated. This results in additional cost, such as crew overtime payments. Delays can cause passengers to miss connecting flights and these passengers have to be rebooked. This also brings additional cost.

However, the impact of a delay will differ from flight to flight, depending, among others, on the number of (transfer) passengers. An airline will often prefer a delay for a flight without any transfer passengers over a delay for a flight full of time-critical transfer passengers. It is expected that by considering these preferences in air traffic control decisions, the impact of delay on the airlines and their passengers can be reduced. This will lead to cost savings for airlines and fewer frustrations for passengers.

Runway operations scheduling involves assigning a landing or take-off time and runway to each flight in such a way that the required separation between flights is respected. The separation required between two flights at the runway depends on the weight categories and sequence of the aircraft. A light aircraft landing behind a heavy aircraft requires more separation than the reverse order. This means that the capacity can be enlarged by actively sequencing the flights. This is import because runways form a major bottleneck of the air traffic system. However, currently flights are not actively sequenced in practice. This means there is an opportunity to improve the efficiency at this bottleneck and with that the efficiency of the total air traffic system.

In this research, possible increases in runway throughput obtained by sequencing the flights are considered. However, the primary objective is to incorporate airline preferences in the runway operation schedule in order to reduce the impact of delays on airline and their passengers. The consideration of both airline preferences and efficiency fills the gap between the two approaches currently considered in the literature to allocate runway capacity to flights.

A novel approach to represent airline preferences and incorporate these in a fair manner in the scheduling process was presented. In this approach, airline preferences are represented using cost functions. These cost functions represents the cost related to runway operations times of flights and connection times between flights. We want to allow the airlines as much flexibility as possible in representing these cost functions. At the same time, these cost functions must be applicable to establish a fair and efficient runway schedule. Therefore, it must be possible to compare the cost functions from competing airlines in a fair manner. Additionally, it should not be possible for airlines to conduct strategic behavior. To achieve this, a combination of centralized decision making and restrictions on the cost functions were proposed. Additional measures of fairness were also defined and evaluated throughout the research.

Two runway operations scheduling problems were studied. First, the single runway aircraft landing problem was considered. Next, the scheduling of arrivals and departures at a hub airport was considered. For both problems, mathematical programming formulations are given and local search heuristics to obtain good solutions using short computation times were introduced. These heuristics has shown to give solutions of good quality for realistically sized instances.

The scheduling of landing flights at a single runway (aircraft landing problem) was tested in computational experiments. For this a large number of problem instances, created using schedule data from a major European hub, were used. The results show tremendous cost savings for the airlines compared to a schedule that resembles current practice, especially at times when runway congestion is expected.

The results also show that schedules with different distributions of cost over the airlines can be obtained, by considering different objectives. There is a trade-off between minimum total cost (over all airlines) and a more equal distribution of cost (savings) and delays over the airlines. However, it was shown that schedules with a more equal cost distribution over the airlines but at the same time considerable total cost savings compared to current practice, can be obtained.

The scheduling of landings and take-offs at multiple runways provided the possibility to explicitly consider hub airline operations. In this way, the costs related to flight connections can be modeled more realistically. Computational experiments for this problem were also performed using data from a large European hub. The results showed that additional cost savings can be obtained by integrally scheduling the runway operations of arrivals and (connected) departures. In this way, for example, the number of missed transfers can be (further) reduced.

We can conclude that the results of our research show that considering airline preferences in runway operations scheduling indeed leads to a reduction of the negative impact of delays to airlines and their passengers. Considerable cost reductions can be obtained for the airlines. Furthermore, passenger frustrations related to delays and missed transfers can be reduced.

Now that the potential gains are established, further research is necessary to allow for the practical application of the approach. In an operational environment runway operation schedules must be calculated almost instantaneously. Fast (real-time) algorithms must be developed to achieve this.

Another interesting subject for future research is whether a similar approach can be used for related air traffic problems, such as air traffic flow control or airport gate assignment.

Bibliography

[1] Coin-or. `http://www.coin-or.org`.

[2] Fliers' misery stings jetblue. USA Today, February 16, 2007.

[3] A. Abdelghany, K. Abdelghany, and G. Ekollu. A genetic algorithm approach for ground delay program management: The airlines' side of the problem. *Air Traffic Control Quarterly*, 12(1):53–74, 2004.

[4] J. Abela, D. Abramson, M. Krishnamoorthy, A. De Silva, and G. Mills. Computing optimal schedules for landing aircraft. In *Proceedings of the 12th National ASOR Conference*, pages 71–90, 1993.

[5] I.A. Anagnostakis and J-P. Clarke. Runway operations planning: A two-stage heuristic algorithm. In *AIAA Aircraft, Technology, Integration and Operations Forum, Los Angeles, CA, October 1st-3rd, 2002*, 2002.

[6] K. Andersson, W. Hall, S. Atkins, and E. Feron. Optimization-based analysis of collaborative airport arrival planning. *Transportation Science*, 37(4):422–433, 2003.

[7] G. Andreatta, L Brunetta, and G. Guastalla. Multi-airport ground holding problem: a heuristic approach based on priority rules. In L. Bianco, P. Dell'Olmo, and A.R. Odoni, editors, *Modeling and Simulation in Air Traffic Management*, pages 71–89. Springer-Verlag, Berlin, 1997.

[8] G. Andreatta, L. Brunetta, and G. Guastalla. From ground holding to free flight: An exact approach. *Transportation Science*, 34(4):394–401, 2000.

[9] K. Artiouchine, P. Baptiste, and C. Dürr. Runway sequencing with holding patterns. *European Journal of Operational Research*, 184(3):1254–1266, 2008.

[10] N. Ashford and P.H. Wright. *Airport Engineering*. Wiley-Interscience, 3rd edition, 1992.

[11] B. Avi-Itzhak and H. Levy. On measuring fairness in queues. *Advances in Applied Probability*, 36:919–936, 2004.

[12] M. Ball, G.L. Donohue, and K. Hoffman. *Combinatorial Auctions*, chapter Auctions for the Safe, Efficient and Equitable Allocation of Airspace System Resources. MIT Press, Cambridge, 2006.

[13] M. O. Ball, R. Hoffman, A.R. Odoni, and R. Rifkin. A stochastic integer program with dual network structure and its application to the ground-holding problem. *Operations Research*, 51(1), 2003.

[14] M.O. Ball, R.L. Hoffman, D. Knorr, J. Wetherly, and M. Wambsganss. Assessing the benefits of collaborative decision making in air traffic management. In *3rd USA/Europe Air Traffic Management R & D Seminar Napoli, 13-16 June 2000*, 2000.

[15] J.B. Barbanel and S.J. Brams. Cake division with minimal cuts: envy-free procedures for three persons, four persons, and beyond. *Mathematical Social Sciences*, 48(3):251–269, 2004.

[16] N. Bäuerle, O. Engelhardt-Funke, and M. Kolonko. On the waiting time of arriving aircrafts and the capacity of airports with one or two runways. *European Journal of Operational Research*, 177(2):1180–1196, 2007.

[17] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha, and D. Abramson. Scheduling aircraft landings - the static case. *Transportation Science*, 34(2):180–197, 2000.

[18] J.E. Beasley, M. Krishnamoorthy, Y.M. Sharaiha, and D. Abramson. Displacement problem and dynamically scheduling aircraft landings. *Journal of the Operational Research Society*, 55(1):54–64, 2004.

[19] J.E. Beasley, J. Sonander, and P. Havelock. Scheduling aircraft landings at London Heathow using a population heuristic. *Journal of the Operational Research Society*, 52(5):483–493, 2001.

[20] R. Beatty, R. Hsy, L. Beny, and J. Rome. Preliminary evaluation of flight delay propagation through an airline schedule. Technical report, FAA CDM Analysis Working Group, 1998.

[21] L. Bianco, P. DellOlmo, and S. Giordania. Minimizing total completion time subject to release dates and sequence-dependent processing times. *Annals of Operations Research*, 86:393–415, 1999.

[22] A. Blumstein. The landing capacity of a runway. *Operations Research*, 7(6):752–763, 1959.

[23] S. Bratu and C. Barnhart. Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, 9(3):279–298, 2006.

[24] P. Brucker, J.K. Lenstra, and A.H.G. Rinnooy Kan. Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1:343–362, 1977.

[25] K. Button. Debunking some common myths about airport hubs. *Journal of Air Transport Management*, 8:177–188, 2002.

[26] G.C. Carr, H. Erzberger, and F. Neuman. Airline arrival prioritization in sequencing and scheduling. In *2nd USA/EUROPE Air Traffic Management R&D Seminar*, 1998.

[27] K. Chang, K. Howard, R. Oiesen, L. Shisler, N. Tanino, and M.C. Wambsganns. Enhancements to the FAA ground delay program under collaborative decision making. *Interfaces*, 31(1):57–76, 2001.

[28] A. Cook, G. Tanner, and S. Anderson. *Evaluating the True Cost to Airlines of One Minute of Airborne or Ground Delay.* EUROCONTROL Performance Review Unit, 2004.

[29] J.I. Daniel. Congestion pricing and capacity of large hub airports: a bottleneck model with stochastic queues. *Econometrica*, 63:327370, 1995.

[30] J.I. Daniel. Distributional consequences of airport congestion pricing. *Journal of Urban Economics*, 50(2):230–258, 2001.

[31] J.I. Daniel and K.T. Harback. (When) do hub airlines internalize their self-imposed congestion delays? *Journal of Urban Economics*, 63:583–612, 2008.

[32] R.G. Dear and Y. S. Sherif. An algorithm for computer assisted sequencing and scheduling of terminal area operations. *Transportation Research - A*, 25(3):129–139, 1991.

[33] J. Du and J.Y.-T. Leung. Minimizing total tardiness on one machine is np-hard. *Mathematics of Operations Research*, 15:483–495, 1990.

[34] A.T. Ernst, M. Krishnamoorthy, and R. H. Storer. Heuristic and exact algorithms for scheduling aircraft landings. *Networks*, 34:229–241, 1999.

[35] H. Erzberger. Design principles and algorithms for automated air traffic management. In *AGARD Lecture Series 200*, 1995.

[36] EUROCONTROL. European CDM portal. `http://www.euro-cdm.org`.

[37] EUROCONTROL. CFMU website. `http://www.cfmu.eurocontrol.int`.

[38] EUROCONTROL. *Central Flow Management Unit Network Operations Report 2007*. EUROCONTROL, 2007.

[39] M. Frank, M. Mederer, B. Stolz, and Hanschke T. Depeaking economic optimization of air traffic systems. *Aerospace Science and Technology*, 9:738–744, 2005.

[40] W.D. Hall. *Efficient Capacity Allocation in a Collaborative Air Transportation System*. PhD thesis, MIT, 1999.

[41] J.M. Hoekstra. Free flight with airborne separation assurance. Technical Report NLR-TP-2002-170, National Aerospace Laboratory NLR, 2002.

[42] R. Hoffman and M.O. Ball. A comparison of formulations for the single-airport ground-holding problem with banking constraints. *Operations Research*, 48(4):578–591, 2000.

[43] R. Hoffman, A. Mukherjee, and T.W.M. Vossen. Air traffic flow management. Working Paper, 2007.

[44] IATA. *Worldwide Scheduling Guidelines*. IATA, 15th edition, 2007.

[45] T. Inniss and M.O. Ball. Estimating one-parameter airport arrival capacity distributions for air traffic flow management. *Air Traffic Control Quarterly*, 12(3):223–251, 2004.

[46] Next Generation Air Transportation (NextGen) Institute. Website. `http://www.ncat.com/ngats/`.

[47] T. Johnson and I. Savage. Departure delays, the pricing of congestion, and expansion proposals at Chicago O'Hare Airport. *Journal of Air Transport Management*, 12:182190, 2006.

[48] B. Kotnyek and O. Richetta. Equitable models for the stochastic ground-holding problem under collaborative decision making. *Transportation Science*, 40(2), 2006.

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

[50] L. Lettovsky, E.L. Johnson, and G.L. Nemhauser. Airline crew recovery. *Transportation Science*, 34(4):337–348, 2000.

[51] L. Kholfi Loan, G. S. Donohue, and C-H. Chen. Proposal for demand management using auction-based arrival slot allocation. In *Proceedings of the 5th EUROCONTROL/FAA ATM R&D Seminar*, 2003.

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

[53] Luchtverkeersleiding Nederland (LVNL). Website (in dutch), uitleg baangebruik (April 2, 2008). `http://www.luchtverkeersleiding.nl/uitleg_baangebruik.html`.

[54] L. Massoulie and J. Roberts. Bandwidth sharing: Objectives and algorithms. *IEEE/ACM Trans. on Networking*, 10(3):320–328, 2004.

[55] A. Mukherjee and M. Hansen. A dynamic stochastic model for the single airport ground holding problem. *Transportation Science*, 41(4), 2007.

[56] H. Pinol and J.E. Beasley. Scatter search and bionomic algorithms for the aircraft landing problem. *European Journal of Operational Research*, 171(2):439–462, 2006.

[57] W. Post and H.W.G de Jonge. Free flight in a ground controlled ATM environment. Technical Report NLR-TP-97594, National Aerospace Laboratory NLR, 1997.

[58] H. N. Psaraftis. A dynamic programming approach to the aircraft sequencing problem. Technical Report R78-4, Flight Transportation Laboratory, MIT, 1978.

[59] H. N. Psaraftis. A dynamic programming approach for sequencing groups of identical jobs. *Operations Research*, 28(6):1347–1359, 1980.

[60] S.J. Rassenti, V.L. Smith, and R.L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13(2):402–417, 1982.

[61] O. Richetta. Optimal algorithms and a remarkably efficient heuristic for the ground-holding problem in air traffic control. *Operations Research*, 43(5):758–770, 1995.

[62] O. Richetta and A.R. Odoni. Solving optimally the static ground-holding policy problem in air traffic control. *Transportation Science*, 27:228–238, 1993.

[63] O. Richetta and A.R. Odoni. Dynamic solution to the ground holding policy problem in air traffic control. *Transportation Reserach Part A*, 28, 1994.

[64] A.H.G. Rinnooy Kan. *Machine Scheduling Problems: Classification, Complexity and Computations.* Martinus Nijhoff, The Hague, The Netherlands, 1976.

[65] G. Sabin, G. Kochhar, and P. Sadayappan. Job fairness in non-preemptive job scheduling. In *International Conference on Parallel Processing, ICPP*, pages 186–194, 2004.

[66] J.L. Schank. Solving airside airport congestion: Why peak runway pricing is not working. *Journal of Air Transport Management*, 11:417–425, 2005.

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

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

[69] M.J. Soomer. Hub airport runway scheduling using airline costs. 2008. Submitted for publication.

[70] M.J. Soomer and G.J. Franx. Equitable airport arrival scheduling using airline costs. In *Proceedings of the 2nd International Conference on Research in Air Transportation (ICRAT)*, 2006.

[71] M.J. Soomer and G.J. Franx. Scheduling aircraft landings using airlines' preferences. *European Journal of Operational Research*, 190(1):277–291, 2008.

[72] M.J. Soomer and G.M. Koole. Fairness in the aircraft landing problem. 2008. Submitted for publication.

[73] M. Stojkovic, F. Soumis, and J. Desrosiers. The operational airline crew scheduling problem. *Transportation Science*, 32(3):232–245, 2000.

[74] R. Stolletz. Non-stationary delay analysis of runway systems. *OR Spectrum*, 30(1):191–213, 2008.

[75] M. Terrab and A.R. Odoni. Strategic flow managment for air traffic control. *Operations Research*, 41(1):138–152, 1993.

[76] T. Vossen and M. Ball. Optimization and mediated bartering models for ground delay programs. *Naval Research Logistics*, 53(1):75–90, 2006.

[77] T. Vossen and M. Ball. Slot trading opportunities in collaborative ground delay programs. *Transportation Science*, 40(1):29–43, 2006.

[78] H.P. Young. *Equity in Theory and Practice.* Princeton University Press, 1994.

Samenvatting

Plannen van operaties op start- en landingsbanen op basis van voorkeuren van luchtvaartmaatschappijen

Een van de grootste irritaties van vliegtuigpassagiers is ongetwijfeld vertraging. Helemaal als een transferpassagier hierdoor een aansluitende vlucht mist. Vaak zal de passagier hierover klagen bij de luchtvaartmaatschappij. Niet bij alle vertragingen ligt de oorzaak (direct) bij de luchtvaartmaatschappij. Een van de belangrijkste reden voor vertraging zijn opstoppingen in het luchtverkeer. Net zoals op de weg, ontstaan er ook in het luchtverkeer files bij grote drukte.

De hoeveelheid luchtverkeer is in de afgelopen decennia enorm toegenomen. De fysieke capaciteit van het luchtverkeerssysteem, waaronder luchthavens en landingsbanen, heeft hier geen gelijke tred mee gehouden. Dit verklaart waarom opstoppingen, en daarmee vertragingen, steeds vaker voorkomen.

Tegelijkertijd hebben veel luchtvaartmaatschappijen het momenteel moeilijk. De operationele kosten zijn hoog, helemaal met de huidige olieprijs. De competitie is hevig. Toeristen zijn, mede door budget luchtvaartmaatschappijen zoals Easyjet en RyanAir, gewend geraakt aan lage ticketprijzen. Dit maakt het noodzakelijk voor luchtvaartmaatschappijen om efficiënt te werken. Bij het nemen van bijna elke beslissing, van het bepalen van de dienstregeling en het inplannen van personeel en vliegtuigen tot het verkopen van stoelen voor de juiste prijs, worden geavanceerde wiskundige optimalisatie modellen gebruikt die helpen om te bepalen hoe dit zo efficiënt mogelijk kan gebeuren.

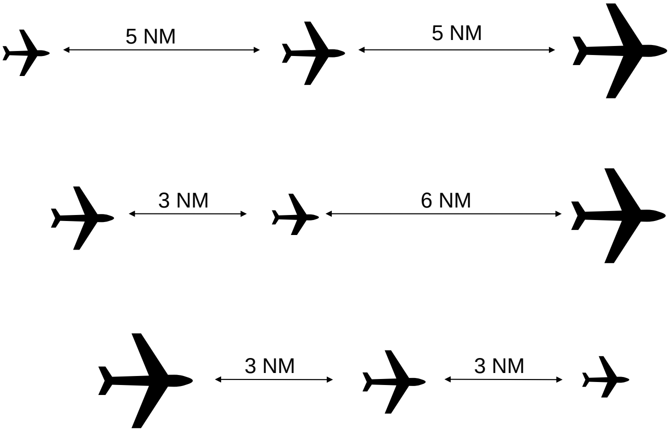
De vertragingen die veroorzaakt worden door opstoppingen in het luchtverkeer verstoren deze (efficiënte) werkwijze, maar liggen grotendeels buiten de invloed van de luchtvaartmaatschappijen. De luchtverkeersleiding is namelijk verantwoordelijk voor het regelen van het luchtverkeer. Deze vertragingen zorgen, naast het ongemak voor de passagiers, echter wel voor extra kosten voor de luchtvaartmaatschappijen. Als een vlucht om moet vliegen of moet wachten om te landen, wordt er meer kerosine verbruikt. Vertraging kan er ook voor zorgen dat de gemaakte planningen voor personeel en vliegtuigtoewijzing voor latere vluchten niet meer uitgevoerd kunnen worden. Daardoor kunnen vertragingen dus doorwerken naar latere vluchten en zo ook voor die vluchten extra kosten veroorzaken.

Door een vertraging kan het ook gebeuren dat een transferpassagier zijn aansluitende vlucht mist. Ook dit leidt tot extra kosten. De passagiers moeten worden omgeboekt naar andere vluchten en krijgen soms een vergoeding voor het ongemak. Bij een groot aantal vertraagde vluchten kan het lastig zijn om alle passagiers te accommoderen. Als er geen latere vlucht meer beschikbaar is, moet er een hotelovernachting voor de passagiers worden geregeld.

Het bovenstaande laat zien dat vertragingen enorme gevolgen hebben voor luchtvaartmaatschappijen en hun passagiers. Deze impact verschilt echter enorm van vlucht tot vlucht. Een belangrijke factor die deze impact bepaald is bijvoorbeeld het aantal (transfer)passagiers op een vlucht. Een luchtvaartmaatschappij zal meestal de voorkeur geven aan een vertraging voor een vlucht zonder transferpassagiers boven een vertraging voor een vlucht vol met transferpassagiers die korte overstaptijden hebben. De verwachting is nu dat als de luchtverkeersleiding rekening houdt met dit soort voorkeuren, de (negatieve) gevolgen van vertragingen voor luchtvaartmaatschappijen en hun passagiers beperkt kunnen worden.

Het doel van dit onderzoek is om te analyseren op welke manier er met de voorkeuren van luchtvaartmaatschappijen rekening kan worden gehouden en wat de gevolgen hier van zijn. Hierbij is specifiek gekeken naar het plannen van vliegtuigbewegingen op start- en landingsbanen.

Bij het plannen van operaties op start- en landingsbanen moet rekening worden gehouden met de afstand die vanwege de veiligheid tussen twee vliegtuigen moet worden bewaard. Deze afstand is afhankelijk van de gewichtsklassen en volgorde van de vliegtuigen. Er is een grotere tussen afstand vereist als een zwaar vliegtuig voor een licht vliegtuig land dan bij de omgekeerde volgorde. In figuur 7.1 worden de vereiste afstanden in zeemijlen (NM) tussen vliegtuigen van verschillende gewichtsklassen voor verschillende volgordes weergegeven.



Figuur 7.1: De benodigde afstand tussen een licht, medium en zwaar vliegtuig hangt af van de landingsvolgorde

De baancapaciteit kan dus vergroot worden door de volgorde van de vliegtuigen op een slimme manier te kiezen. Dit is belangrijk omdat start- en landingbanen een belangrijke bottleneck in het luchtverkeerssysteem vormen. In de praktijk wordt de volgorde van de vliegtuigen echter niet geoptimaliseerd. Hier ligt dus een kans om de efficiency van deze bottleneck te vergoten en daarmee de efficiency van het gehele luchtverkeerssysteem.

In dit onderzoek worden de mogelijkheden om de capaciteit optimaal te gebruiken door het aanpassen van de volgorde ook in beschouwing genomen. Het belangrijkste doel is echter om de voorkeuren van luchtvaartmaatschappijen te verwerken in de planning.

Een complicatie bij het verwerken van de voorkeuren van luchtvaartmaatschappijen is dat ook de eerlijkheid in de gaten moet worden gehouden. Het is de taak van de luchtverkeersleiding om het luchtverkeer op een veilige, efficiënte en eerlijke manier te leiden. Schaarse baancapaciteit moet daarom op een eerlijke manier over concurrerende luchtvaartmaatschappijen worden verdeeld.

In het onderzoek wordt een nieuwe aanpak voorgesteld om voorkeuren van luchtvaartmaatschappijen op een eerlijke manier te kunnen gebruiken bij het plannen. De voorkeuren van de luchtvaartmaatschappijen worden weergegeven door kostenfuncties. Deze kostenfuncties representeren de relatie tussen de kosten van een luchtvaartmaatschappij en het plannen van een vlucht op een bepaald tijdstip of de relatie tussen de kosten en het tijdsverschil tussen twee vluchten. Het is de bedoeling om de luchtvaartmaatschappijen zoveel mogelijk vrijheid te geven bij het vaststellen van deze kostenfuncties. Aan de andere kant, moet het wel mogelijk zijn om de kostenfuncties te gebruiken om een eerlijke en efficiënte planning te bepalen. Hiervoor moet het mogelijk zijn om de kostenfuncties van concurrerende luchtvaartmaatschappijen op een eerlijke manier te vergelijken. Ook is het niet wenselijk als luchtvaartmaatschappijen strategisch gedrag vertonen. Om deze doelen te bereiken wordt er in het onderzoek een aanpak voorgesteld die bestaat uit een aantal voorwaarden met betrekking tot de kostenfuncties die luchtvaartmaatschappijen mogen gebruiken en uit het centraal bepalen van een planning. In het onderzoek worden er ook aanvullende criteria voor eerlijkheid gedefinieerd en geëvalueerd.

In het onderzoek zijn er twee type problemen met betrekking tot het plannen van operaties op start- en landingsbanen bekeken. Het eerste type probleem dat is het plannen van landingen op één landingbaan. Het tweede type probleem is het plannen van zowel landingen als starts op meerdere start- en landingsbanen. Voor beide problemen zijn er wiskundige modellen en heuristieken opgesteld.

Deze modellen zijn doorgerekend met behulp van data van een grote Europese luchthaven. Uit deze berekeningen blijkt dat er met deze aanpak grote kostenbesparingen voor luchtvaartmaatschappijen mogelijk zijn ten opzichte van de methode die nu in de praktijk wordt gebruikt. Dit is vooral het geval in drukke periodes, waarin oponthoud wordt verwacht.

Bij het tegelijkertijd plannen van landingen en starts kan op een realistische manier rekening worden gehouden met de kosten die gerelateerd zijn aan transfers. De resultaten laten zien dat er op deze manier additionele kostenbesparingen bereikt kunnen worden. Op deze manier kan ook het aantal gemiste transfers worden beperkt.

Samengevat kunnen we concluderen dat door rekening te houden met de voorkeuren van luchtvaartmaatschappijen bij het plannen van operaties op start- en landingsbanen, de negatieve gevolgen van vertragingen voor luchtvaartmaatschappijen en hun passagiers inderdaad beperkt kunnen worden.

About the Author

Maarten Soomer was born at May 30, 1978 in Amstelveen.

In 2003 he received his master in Business Mathematics and Informatics at the VU University Amsterdam. His internship was performed at the National Aerospace Laboratory NLR on the subject of optimization of airport taxi planning.

After his graduation, he became a PhD student at the VU University Amsterdam and the National Aerospace Laboratory NLR.

In 2007 he took a 6 months break from his PhD research to work on a Revenue Management project at Park ’N Fly in Atlanta, USA. This revitalized his interest in the area of Revenue Management and in August 2008 he started working as consultant in this area at ORTEC.

Maarten defends his PhD thesis at the VU University Amsterdam on January 9, 2009.