Comparing airport apron layout designs using computer simulation and the cross-entropy method

Trudie Leonard

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in the Faculty of Engineering at Stellenbosch University

Study leader: J. Bekker

December 2011
Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: ................................................

Copyright © 2011 Stellenbosch University
All rights reserved
Abstract

The demand in air travel is continuously increasing. In order to handle this increase in demand, airports need to physically expand or the management of the airports needs to improve. When the demand at OR Tambo International Airport gets too high, more passengers will need to travel to Lanseria International Airport, which will therefore need to be expanded. The study was done in collaboration with Virtual Consulting Engineers, who decided that the concept of Atlanta International Airport in Georgia, USA, which is ranked the busiest airport in the world, will be used in this expansion. The aim of the study was to minimise passenger walking distances and waiting times at Lanseria International Airport. This was done by comparing different airport apron layouts, using simulation, and improving the aircraft gate assignment, using the cross-entropy method.

Four different designs of airport layouts, all based on that of Atlanta International Airport, were compared in the study. A model of each was developed using simulation. The performance measures used to compare the designs included 1) the average walking distance of arriving and departing passengers at the airport, 2) the average time spent at the airport by arriving and departing passengers, 3) the average distance travelled by aircraft at the airport, 4) the average time by which each aircraft is delayed and 5) the average number of aircraft present at the airport.

The walking distance of arriving and departing passengers was largely affected by the way in which flights were assigned to gates. The gates at the airport are of three different sizes: small, medium and large. Small aircraft can park at any of the gates, while medium aircraft can only park at medium or large gates and large aircraft can only park
at large gates. Three rules for the flight-to-gate assignment process were developed. In the first two rules an arriving flight was assigned to the available, suitable gate closest to the terminal building. The constraint that small aircraft cannot be assigned to medium or large gates if there are small gates available and that medium aircraft cannot be assigned to large gates if there are medium gates available, was used in Rule 1 and not in Rule 2. In the third rule, metaheuristic optimisation was used to determine a flight-to-gate assignment schedule with the objective of minimising the passenger walking distances. This metaheuristic optimisation was performed in real-time and was thus repeated every time a delay occurred at the airport.

The background of airports, simulation, metaheuristics and relevant case studies were investigated in the literature review. The simulation and metaheuristic optimisation models were then developed. The results identified the best of the four designs that were compared. It was also concluded that the use of metaheuristic optimisation, using the cross-entropy method, results in a reduction in passenger walking distances at the airport.
Opsomming

Die aantal lugpassasiers neem aanhoudend toe en om in staat te wees om hierdie toename in vraag te hanteer moet lughawens fisies uitbrei of die bestuur van die lughawens moet verbeter. Wanneer die vraag by OR Tambo Internasionale Lughawe te hoog raak, gaan meer mense na Lanseria Internasionale Lughawe moet reis. Die lughawe sal dan dus moet uitbrei. Die studie is in samewerking met Virtual Consulting Engineers gedoen. Hulle het besluit dat die konsep van Atlanta Internasionale Lughawe in Georgia in die VSA, wat die besigste lughawe in die wêreld is, gebruik sal word in die uitbreiding Lanseria Internasionale Lughawe. Die doelwit van die studie was om die loopafstand en die wagtyd van passasiers op Lanseria Internasionale Lughawe te minimeer. Die doelwit is bereik deur verskillende lughawe uitlegte te vergelyk met behulp van simulasi en deur die toekenning van vlugte aan hekke te verbeter, deur gebruik te maak van die “cross-entropy” metode.

Die konsep van Atlanta Internasionale Lughawe is gebruik om vier verskillende lughawe uitlegte te ontwerp. Simulasie is gebruik om die vier ontwerpe te vergelyk op grond van 1) die gemiddelde loopafstand van passasiers wat aankom en vertrek, 2) die gemiddelde tyd wat passasiers wat aankom en vertrek spandeer op die lughawe, 3) die gemiddelde afstand wat vliegtuie aflê op die lughawe, 4) die gemiddelde tyd wat vliegtuie vertraag word, 5) die gemiddelde hoeveelheid vliegtuie teenwoordig op die lughawe. Die loopafstand van passasiers wat aankom en vertrek is grootliks beïnvloed deur die manier waarop vliegtuie aan hekke toegeken is. Die hekke op die lughawe is klein, medium of groot. ’n Klein vliegtuig mag by ’n klein, medium of groot hek parkeer, ’n medium vliegtuig
mag by ’n medium of groot hek parkeer en ’n groot vliegtuig mag net by ’n groot hek parkeer. Drie reëls waarvolgens vliegtuie aan hekke toegeken kan word is ontwikkel. In die eerste twee reëls word ’n vliegtuig wat aankom aan die beskikbare hek naaste aan die terminaal gebou toegeken as die hek geskik is vir die vliegtuig. In die eerste reël is die beperking dat klein vliegtuie nie aan medium en groot hekke toegeken mag word as daar klein hekke beskikbaar is nie en dat medium vliegtuie nie aan groot hekke toegeken mag word as daar medium hekke beskikbaar is nie, ingesluit. Hierdie beperking is nie in die tweede reël ingesluit nie. In die derde reël is metaheuristiek optimering gebruik om vliegtuie aan hekke toe te ken. Die doelwit van die metaheuristiek optimering was om die loopafstand van die passasiers te verminder. Elke keer as ’n vliegtuig op die lughawe vertraag was, is die optimering proses herhaal.

Die agtergrond van lughawens, simulasie, metaheuristieke en gevalle studies is bestudeer in die literatuur studie. Daarna is die simulasie en metaheuristiek optimering modelle ontwikkel. Die resultate van die studie het aangedui watter een van die vier lughawe ontwerpe die beste is. Dit is ook beslis dat die gebruik van metaheuristiek optimering, en spesifiek die “cross-entropy” metode, die loopafstand van passasiers op die lughawe verminder.
Contents

Declaration i
Abstract ii
Opsomming iv
Contents vi
List of figures ix

1 Introduction 1
   1.1 Background of the problem .................................... 1
   1.2 Project methodology ............................................. 3
   1.3 Chapter overview of the study .................................. 5
      1.3.1 Background of an airport ................................. 5
      1.3.2 The overview of the experiments ......................... 6
      1.3.3 The problem solving phase .............................. 6
      1.3.4 The results ............................................... 6
      1.3.5 The conclusion ........................................... 7

2 Airport operations 8
   2.1 Demand at an airport ........................................... 10
   2.2 Operating authorities at an airport ......................... 12
   2.3 Passenger handling ............................................ 13
   2.4 Aircraft handling ............................................. 14
   2.5 Concluding remarks on chapter 2 ........................... 15
## CONTENTS

3 Airport components ............................................ 16
   3.1 Airport terminals ........................................... 17
   3.2 Airport aprons ............................................. 18
      3.2.1 Aircraft parking ...................................... 19
      3.2.2 Gate using ........................................... 22
   3.3 Airport runways ........................................... 23
   3.4 Concluding remarks on chapter 3 ......................... 24

4 Airport capacity and conflicts ................................. 25
   4.1 Airport capacity ........................................... 26
   4.2 Conflicts at an airport .................................... 27
      4.2.1 Conflicts due to runway crossings .................... 28
      4.2.2 Conflicts between passengers, vehicles and aircraft 30
   4.3 Concluding remarks on chapter 4 ......................... 31

5 Overview of the experiments in the study ..................... 32
   5.1 Atlanta International Airport .............................. 33
   5.2 Different airport designs to be compared ................. 35
   5.3 Flight to gate assignment in the study ................... 39
   5.4 Considerations in the study ............................... 40
   5.5 Concluding remarks on chapter 5 ......................... 42

6 Simulation as a problem solving technique .................... 43
   6.1 The application of simulation ............................. 44
   6.2 Advantages and disadvantages of simulation ............... 45
   6.3 Components of a simulation model ......................... 46
      6.3.1 Inputs, outputs and states ............................ 46
      6.3.2 Entities and attributes ............................... 47
      6.3.3 Activities and events .................................. 48
      6.3.4 Resources ............................................. 48
      6.3.5 Statistical collectors ................................. 48
   6.4 Steps in a simulation study ................................ 49
   6.5 Concluding remarks on chapter 6 ......................... 49
## CONTENTS

### 7 The simulation models

- 7.1 The entities in the model ----------------------------------------- 52
  - 7.1.1 The aircraft ----------------------------------------------- 52
  - 7.1.2 The arriving passengers ------------------------------------ 56
  - 7.1.3 The departing passengers ------------------------------------ 56
- 7.2 The taxiways in the model --------------------------------------- 58
  - 7.2.1 Entering a taxiway from a gate ------------------------------- 59
  - 7.2.2 Entering a taxiway from the side ----------------------------- 65
  - 7.2.3 Exiting a taxiway ------------------------------------------- 68
- 7.3 The speed at which the aircraft travel ----------------------------- 71
- 7.4 Loading and unloading of passengers ------------------------------- 71
- 7.5 Passenger paths in the concourses ------------------------------- 72
- 7.6 Concluding remarks on chapter 7 --------------------------------- 72

### 8 Validation and verification of the simulation models

- 8.1 Validation and verification techniques used ---------------------- 76
  - 8.1.1 Model reasonableness --------------------------------------- 78
  - 8.1.2 Face validation --------------------------------------------- 79
- 8.2 Concluding remarks on chapter 8 -------------------------------- 81

### 9 Objective optimisation by using the cross-entropy method

- 9.1 Different metaheuristic techniques ------------------------------- 84
- 9.2 The cross-entropy method ---------------------------------------- 86
  - 9.2.1 Steps and procedures in the CE method ----------------------- 87
  - 9.2.2 Mathematical formulation of the CE method ------------------- 88
- 9.3 Concluding remarks on chapter 9 --------------------------------- 91

### 10 Gate assignment operation case studies

- 10.1 A case study by Yan et al. ------------------------------------- 93
- 10.2 A case study by Drexel and Nikulin ----------------------------- 97
- 10.3 Concluding remarks on chapter 10 ------------------------------- 101
## CONTENTS

11 Assigning flights to gates using the cross-entropy method 102
   11.1 Overview of the optimisation process .......................... 105
   11.2 Frequency of performing the optimisation process .......... 107
   11.3 Actions performed only once .................................. 110
      11.3.1 Selecting the first arriving flight in the run .......... 111
      11.3.2 Selecting the first departing flight in the run ....... 113
      11.3.3 Assigning the last flight in the run ................. 114
      11.3.4 Test for available gates ................................ 116
   11.4 Creating the population ....................................... 117
      11.4.1 Releasing the gates when the flights depart ....... 119
      11.4.2 Occupying the gates when the flights arrive ...... 121
      11.4.3 Calculating the performance of each solution ...... 125
   11.5 The application of the CE method ............................. 126
   11.6 The generic model of the optimisation process using the CE method 129
   11.7 This study versus the case studies ............................ 130
   11.8 Concluding remarks chapter 11 ............................... 131

12 Data and results 132
   12.1 The data used in the models .................................. 133
   12.2 The results of the simulation models ......................... 133
      12.2.1 The statistics used to compare the scenarios .......... 134
      12.2.2 Presentation of the results .............................. 135
      12.2.3 Interpretation of the results ............................ 144
      12.2.4 The busiest schedule to be used by the airport .... 148
   12.3 Concluding remarks on chapter 12 ........................... 149

13 Conclusion 150

References 159
List of Figures

1.1 Project layout ................................................................. 5
3.1 Passenger walking distance ............................................. 19
3.2 Aircraft parking: nose-in ................................................. 20
3.3 Aircraft parking: angled nose-in ....................................... 21
3.4 Aircraft parking: angled nose-out ..................................... 21
3.5 Aircraft parking: parallel ................................................ 22
3.6 Aircraft parking: remote ................................................ 22
5.1 Airport apron layout: Design 1 ......................................... 36
5.2 Airport apron layout: Design 2 ......................................... 37
5.3 Airport apron layout: Design 3 ......................................... 38
5.4 Airport apron layout: Design 4 ......................................... 39
7.1 Taxiways ................................................................. 58
7.2 Taxiway with gates ...................................................... 59
7.3 Entering taxiway from a gate, scenario 1 .............................. 60
7.4 Entering taxiway from a gate, scenario 2 .............................. 60
7.5 Entering taxiway from a gate, scenario 3 .............................. 61
7.6 Entering taxiway from a gate, scenario 4 .............................. 62
7.7 Entering taxiway from a gate, scenario 5 .............................. 62
7.8 Updating aircraft places in the line ................................... 64
7.9 Entering taxiway from the side, scenario 1 ............................ 66
7.10 Entering taxiway from the side, scenario 2 ............................. 66
7.11 Entering taxiway from the side, scenario 3 ............................. 67
7.12 Entering taxiway from the side, scenario 4 ............................. 67
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.13</td>
<td>Entering taxiway from the side, scenario 5</td>
<td>68</td>
</tr>
<tr>
<td>7.14</td>
<td>Exiting a taxiway</td>
<td>69</td>
</tr>
<tr>
<td>7.15</td>
<td>Concept model</td>
<td>74</td>
</tr>
<tr>
<td>11.1</td>
<td>The execution of the optimisation process</td>
<td>104</td>
</tr>
<tr>
<td>11.2</td>
<td>Schematic presentation of the optimisation process</td>
<td>106</td>
</tr>
<tr>
<td>11.3</td>
<td>The first arriving flight to consider in the optimisation process</td>
<td>112</td>
</tr>
<tr>
<td>11.4</td>
<td>The first departing flight to consider in the optimisation process</td>
<td>114</td>
</tr>
<tr>
<td>11.5</td>
<td>Execution of the optimisation process</td>
<td>115</td>
</tr>
<tr>
<td>11.6</td>
<td>Execution of the optimisation process with delays</td>
<td>115</td>
</tr>
<tr>
<td>11.7</td>
<td>The empirical distribution of the probability of flight $i$ parking at gate $j$</td>
<td>123</td>
</tr>
<tr>
<td>12.1</td>
<td>Selected airport apron layout: Design 2</td>
<td>148</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This chapter provides an overview of the problem which is solved in this study. A short background is given, the objectives of the study are listed and the layout of the document in terms of the contents of each chapter is discussed.

1.1 Background of the problem

The study was done in collaboration with Virtual Consulting Engineers (VCE) in Pretoria. When the demand in aircraft and passenger traffic gets too high at OR Tambo International Airport, another airport will need to be built or an existing one will need to be expanded. After extensive studies done by VCE, it was concluded that the best option will be to expand Lanseria International Airport. The concept that will be used in the expansion of this airport, is that of Hartsfield Jackson Atlanta International Airport in Georgia, USA. This airport is ranked the busiest airport in the world and will be referred to as Atlanta Airport in this study.

The concept of Atlanta Airport was used to design four different airport layouts. This airport consists of a terminal building and five concourses. The boarding gates are on either side of each of these concourses. Passengers check in at the terminal building and then move to the concourses from which their flights are departing. In the first design, the exact layout of Atlanta Airport is used. The only differences between this design and that of Atlanta are the number of gates...
1.1 Background of the problem

and the dimensions of the apron. In the other three designs, the concept of Atlanta Airport is also used, but the orientation of the terminal with respect to the concourses is changed. At Atlanta Airport, the terminal and all the concourses are connected via an underground transportation mall. Automatic people movers (small trains) transport the passengers from the terminal to the concourses and back, and from one concourse to another. However, at the new Lanseria Airport, passengers will not be transported by automatic people movers, but by pedestrian walkways (similar to conveyor belts) or by foot. This results in the problem of long passenger walking distances at the airport. The study is thus concerned with reducing the total passenger walking distance at the airport.

The research aim is formulated as follows:

Minimise the passenger walking distances and waiting times at an airport by improving the apron layout and the aircraft gate assignment process.

The following objectives were identified for the study:

1. Develop different apron layouts.

2. Develop gate assignment rules/algorithms for arriving aircraft.

3. Develop, as one of the gate assignment rules, a generic optimisation process, using metaheuristic optimisation and specifically the cross-entropy method, which can be used in any airport model to improve the passenger walking distances.

4. Evaluate the different airport layout designs using computer simulation, while considering the following performance measures: 1) the average walking distance of arriving and departing passengers at the airport, 2) the average time spent at the airport by arriving and departing passengers, 3) the average distance travelled by aircraft at the airport, 4) the average time by which each aircraft is delayed and 5) the average number of aircraft present at the airport.
1.2 Project methodology

5. Determine whether the use of metaheuristic optimisation, and specifically the cross-entropy method (thus the use of the optimisation process as stipulated in objective 3) to assign flights to gates, results in a decrease in passenger walking distances at an airport.

6. Determine whether the cross-entropy method can be applied in real-time to simulation problems. The application of the cross-entropy method in this study is dynamic in two ways: 1) the simulation problem is dynamic in the sense that the state of the airport changes over time, and 2) the cross-entropy method is applied in real-time to account for delays.

The project methodology will be developed in a way that will result in meeting these objectives. The methodology is discussed in the next section.

1.2 Project methodology

Four different apron designs will be developed. A simulation model of each of the designs will then be built and each design will be evaluated based on the aforementioned performance measures in order to determine which airport layout should be used in the expansion of Lanseria International Airport. Factors that will be considered in the design of the simulation models include the following:

- the schedule of flight arrivals and departures at the airport
- the layout of the apron
- the travelling speed of the aircraft on the different types of taxiways
- the walking speed of passengers when walking freely and on the pedestrian walkways (see section 2.3)
- the time it takes to load and unload passengers
- the avoidance of collisions between aircraft on the taxiways
- the gate to which each aircraft is assigned on arrival
1.2 Project methodology

The last mentioned factor, deciding to which gate to assign each arriving flight, contributes to the passenger walking distance and will be performed based on three different rules in this study. In the first rule, an arriving flight will be assigned to the available, suitable gate closest to the terminal building. However, in this rule, a small aircraft cannot be assigned to a medium or large gate if there are small gates available, and a medium aircraft cannot be assigned to a large gate if there are medium gates available. In the second rule, an arriving flight will also be assigned to the available, suitable gate closest to the terminal building, but now small aircraft can be assigned to medium or large gates even though there are small gates available and medium aircraft can be assigned to large gates even though there are medium gates available. In the third rule, metaheuristic optimisation will be used to assign arriving flights to gates with the objective of reducing the passenger walking distance for all the flights. Thus, instead of assigning an arriving flight to the gate that is best for this flight in terms of passenger walking distance, in Rule 3 the flight will be assigned to a gate that is best for this flight and a number of flights arriving after it.

The cross-entropy method, that will be explained in chapter 9, will be used for the metaheuristic optimisation in Rule 3. This optimisation process will be performed in real time and will therefore be repeated every time a delay occurs at the airport. If no delays occur, the metaheuristic optimisation will be performed on the arrival of every 25th flight. The current arriving flight and the 50 flights arriving after this flight will be considered each time the process is executed. In each execution, a number of flight-to-gate assignment combinations will be created and evaluated based on the passenger walking distance in each. The good combinations will then be used to determine the way in which to assign the flights to gates in the next iteration of the current optimisation process execution. This will be repeated until the total passenger walking distance is satisfactory. The best calculated flight-to-gate assignment combination in the final iteration will then be used in the simulation. When a flight arrives, it will start taxiing to the gate to which it was assigned in the most recent execution of the metaheuristic optimisation process.

The following factors will be considered in the metaheuristic: 1) each flight may only be assigned to one gate, 2) only one flight may assigned to a gate at
any time. Thus, a flight may only be assigned to a gate if the flight that was previously assigned to that gate has departed and 3) a large aircraft may not be assigned to a small gate.

### 1.3 Chapter overview of the study

The layout of the project is illustrated in figure 1.1. The study can be divided into the different parts represented in the figure. These different parts are discussed in sections 1.3.1 to 1.3.4.

**Figure 1.1: Project layout**

#### 1.3.1 Background of an airport

In chapter 2 the operations at an airport are described. Specific focus is placed on the users, the operating authority and aircraft and passenger handling at the airport. Chapter 3 provides an overview of airport terminals, aprons and
1.3 Chapter overview of the study

runways. In chapter 4, the capacity of an airport as well as conflicts between aircraft, passengers and other vehicles are discussed.

1.3.2 The overview of the experiments

In chapter 5 an overview of the experiments in this study is provided and the different airport designs are shown. The three rules for assigning flights to gates and the differences between these rules are discussed. Also, considerations made in the study are pointed out in this chapter.

1.3.3 The problem solving phase

The problem solving phase of this study can be divided into two parts:

- the simulation models
- the optimisation process

In chapter 6 simulation as a problem solving technique is discussed by explaining the different components in a simulation study and in chapter 7 the logic in the simulation models built for the purpose of this study is presented. In chapter 8 verification and validation of the simulation models are performed.

In chapters 9 and 10 different objective optimisation methods and metaheuristic techniques are explained and two case studies in this regard are presented. Chapter 11 encompasses the logic behind the metaheuristic optimisation process developed for flight-to-gate assignments in the simulation models in order to reduce passenger walking distances.

1.3.4 The results

The results obtained from the simulation models of the different designs, using Rule 1, 2 and 3 of assigning flights to gates, are presented in chapter 12. Furthermore, the data used in the simulation models and the adjustment of the data to suit the airport designs are discussed in this chapter.
1.3.5 The conclusion

In chapter 13, the literature review, the simulation study and the metaheuristic optimisation, are summarised. The most important results obtained in the study are also summarised in this chapter. Conclusions are drawn and the way in which the objectives of the study were met is discussed.
Chapter 2

Airport operations

According Ashford et al. (1997), the function of an airport is to be either the starting, intermediate or final point of an airborne trip. The airport must thus be designed so that it can handle take-offs and landings of aircraft, loading and unloading of passengers, luggage and crew and servicing of the aircraft. The operations of an airport are divided into airside and landside operations. This can be illustrated by dividing the airport between the gates that lead to the aircraft and the side of the airport where passengers are checked-in and security checks
are done. The airside is mainly used by aircraft in that it includes the runways, taxiways and aprons. The passengers disembark the aircraft at the gates and as they move through the gates, they enter the landside, where they get their luggage. This is the process for arriving passengers. For departing passengers, the process is reversed and passengers start in the landside area to check in and then move through the gates to the airside area where they get onto the aircraft. In this chapter, the functions and operation of an airport are discussed while considering the airport users. Also, the difference between centralised and decentralised airports is pointed out.

An airport must be able to do the following (Ashford et al., 1997):

- handle passengers
- service and maintain aircraft
- efficiently manage ground crew, air crew and other staff
- accommodate business necessary for economic stability
- control air traffic
- administer government functions such as inspections, customs, health and immigration

Ground handling activities can be divided into terminal and airside activities. The ground handling activities in the terminal include (Ashford et al., 1997):

- luggage check
- luggage handling
- luggage claim
- check-in and ticketing
- loading and unloading of passengers
- handling of transit passengers
- handling of passengers that need special assistance
2.1 Demand at an airport

- information systems
- government controls
- control of load
- security checks
- cargo

Ground handling activities on the airside include (Ashford et al., 1997):

- ramp services such as supervision, start up, marshalling, towing of aircraft, and measures to ensure safety
- aircraft servicing on the ramps which include repairs, refuel, wheel and tire checks, power supply on ground, heating or cooling, servicing of the toilets, portable water, general maintenance, non-general maintenance and outside cleaning
- onboard services such as cleaning, catering, in-flight entertainment and servicing of cabin fittings
- equipment on external ramps, like passenger steps, loaders for catering, cargo, mail and crew steps

Many stakeholders and users are involved at an airport, as will be discussed in the next section. The above mentioned activities must therefore be managed efficiently in order to satisfy these stakeholders.

### 2.1 Demand at an airport

The users of an airport include aircraft, passengers, cargo and surface vehicles. The aircraft are accommodated on the airside while the passengers, surface vehicles and cargo are accommodated on the landside. The airside can be divided into the airfield, which accommodates all the facilities on ground, and the airspace, which is the off-the-ground area that surrounds the airport. The landside can
be divided into the terminals facilitating passengers and cargo, and the ground access area facilitating the movement of surface vehicles (Wells & Young, 2004).

Air traffic is continuously increasing and a lot of attention has been given by research in air traffic management to the efficiency of arrivals and departures of aircraft. Since these arrivals and departures take place at the airport, ground delays and taxi efficiency are becoming more evident. Thus, a more efficient air transport system is required to handle the increasing demand in air travel (Cheng et al., 2001). According to Offerman (2001), an average increase of 7–10% in flight movements occurs at airports each year. This results in serious bottlenecks in terms of passenger capacity. Most of the times, since the number of runways, ramps and taxiways already reached the limit, the airport cannot be physically expanded as the demand increases. Furthermore, airports are often restricted in terms of environmental safety which may cause a further burden in the airport operations. An airport must therefore be able to grow in a sustainable way.

As stated in the article of Zografos & Madas (2006), airport design, planning and operations come together with a lot of complex problems with regard to decision-making. These problems involve strategic planning, operations management, a wide variety of entities (like passengers, cargo, aircraft and luggage) that must be managed, and elements such as the runways, taxiways, terminals and aprons that must be operated. Furthermore, many stakeholders are involved in an airport system that all have their own, often conflicting, objectives. Decision makers therefore need a way of evaluating all indicators of the effectiveness of the airport while considering their trade-offs (Offerman, 2001).

Issues that the stakeholders of an airport may have are listed below (Offerman, 2001):

- delays as well as arrival and departure punctuality
- capacity in terms of throughput
- slot coordination and allocation
- the robustness of the timetable
- safety of passengers and general public
2.2 Operating authorities at an airport

- noise contamination
- the departure and arrival routes

Simulation is an appropriate technique to satisfy all stakeholders and optimise their conflicting objectives, since future concepts can be tested, timetable feasibility can be evaluated, different runway configurations can be compared and bottlenecks can be identified (Offerman, 2001). Economic efficiency of an airport is measured by indicators such as revenues, operating ratio and return on investment. Customer satisfaction is measured based on waiting time, delays, walking distances, number of accidents and complaints. These are incorporated into one or more levels of service (LOS) (Caves & Gosling, 1999).

2.2 Operating authorities at an airport

The layout of the airport affects the operating authority. Terminal systems can either be centralised or decentralised. In the beginning, when the air transport industry was very small, the centralised concept was used in most airports. In this concept, all passenger- and other processes are carried out in the main terminal building. This building is then connected to the gates by piers or transporters. Brussels airport still uses this concept. Airports such as London Heathrow and OR Tambo in Johannesburg started by using the centralised concept, but as the traffic increased, terminals were added and these airports started to operate in a decentralised manner. Other airports were decentralised from the beginning where a number of terminals, each with a complete set of facilities, exist. These include Paris Charles de Gaulle and New York JFK. Atlanta International Airport uses decentralisation with extensive remote pier developments as will be explained in chapter 5 (Ashford et al., 1997).

Airports of which the physical size started as small, centralised facilities, but experienced large increases in demand had to be expanded. The parking areas also had to be expanded. Those airports that stayed centralised now had very long walking distances since, no matter where they parked and at which gates their flights arrived, passengers had to walk to the main terminal building. To overcome
2.3 Passenger handling

the problem of unsatisfactory walking distances, airports became decentralised (Ashford et al., 1997).

Other advantages of decentralisation are that terminals do not become unmanageably large and that passenger volumes at a single terminal do not become uncomfortably high. Parking lots also stay small. However, more staff is needed at decentralised airports, since the same functions must now be carried out separately at each terminal. Interlining and transferring passengers must have a way to move between terminals, in the case of decentralisation. Some airports use automatic transit vehicles for this purpose, while others use bus services. However, neither of these are very convenient for the passengers (Ashford et al., 1997).

2.3 Passenger handling

Except for government controls such as health, immigration and customs, passenger handling is almost entirely the responsibility of the airlines and not the airport. However, some airports use common user terminal equipment (CUTE) where check-in counters are used for all airlines, with the check-in clerk connected to the airline computers instead of using specific check-in counters for only one airline. This reduces the required number of check-in counters especially where there are many airlines, where airline presence is not required throughout the day or where the schedules of some airlines are very light. Airlines vacate the check-in counters when their departure process is finished and the counters are then occupied by the next airline.

Frequently, the loading bridges as well as the transfer passenger steps are operated by the airlines. Usually, passengers are moved on the apron via bus, of which both airport and airline take ownership and handle operations (Ashford et al., 1997).

Within the terminal, passengers must be processed with regard to ticketing, check-in, luggage drop and claim, and government and security checks. The airport must be able to arrange passengers arriving via different ways of transport and from different access roads, into plane loads for departure. This is also done in the terminal building. This process is carried out in reverse for arriving passengers. The terminal operations include managing the interface between the
2.4 Aircraft handling

airside and the landside for smooth transferring of passengers from one to the other (Ashford et al., 1997).

In the case of decentralised airports, it may be necessary for passengers to move from one terminal to another. These terminals are, however, often spaced very far from each other which makes walking between them inconvenient or even impossible for passengers. The following three methods for moving passengers have been developed (Ashford et al., 1997):

- buses
- pedestrian walkways (operated like conveyor belts)
- automatic people movers (as are used at Atlanta International Airport)

The main limitation of pedestrian walkways is the speed of movement that must be kept below 2.5 km/h. They can therefore not be used for very long distances. Furthermore, if the walkway fails, walking may be the only other option. Automatic People movers can move people at up to 45 km/h. For these systems, however, stations, tracks, control rooms, areas for maintenance, emergency areas and escape points must be provided. Also, in the case of a failure, alternative methods of travel must be available. Atlanta airport is a large hub airport and thus has to be able to handle a lot of transfer passengers. These passengers will often need to move between terminals. An efficient method of travel must thus be used (Ashford et al., 1997).

2.4 Aircraft handling

While aircraft are on the ground, whether in transit, turn-around or parking stage, the apron has to accommodate a lot of activity. After arrival, the aircraft is guided by a marshal to go through all procedures to park safely and in the right position.

Before departure, ramp handling includes going through all the required procedures for take-off. Ramp handling may also include towing of aircraft if it needs to be moved to a different location (Ashford et al., 1997).
2.5 Concluding remarks on chapter 2

The ramp servicing process includes a large number of activities. Unless those activities can be carried out simultaneously, the turnaround time for the aircraft will be too long. The aircraft mobile equipment and apron must thus be designed in a way that will allow the activities to be carried out efficiently (Ashford et al., 1997).

2.5 Concluding remarks on chapter 2

From the discussion in this chapter, based on the literature studied, it is clear that there are a lot of activities and functions that must be performed at an airport. These functions must be operated in a way that will satisfy all stakeholders involved at the airport. Flight delays must be avoided and passenger walking distances must be kept low. The operating authority at the airport, i.e. whether the airport is centralised or decentralised, plays an important role in the efficiency of the airport. In chapter 3 the different components of an airport will be discussed while specifically focusing on the airport terminals, aprons and runways. Factors discussed in chapter 2 such as centralised and decentralised airports, aircraft handling and passenger handling (specifically the passenger walking distances) will be considered in chapter 3.
Chapter 3

Airport components

In this chapter, the different components of an airport are discussed. The three main components of an airport are:

- the terminal/terminals
- the apron
- the runways
3.1 Airport terminals

The following sections elaborate on these components. Different terminal concepts are compared, the components of an apron are discussed and the operation of airport runways is explained together with a discussion on runway capacity.

3.1 Airport terminals

The terminals at an airport are used to process passengers, crew and cargo and facilitate their movement on and off the aircraft. They are, however, not starting- and end points for passengers and cargo, but they serve as transfer areas (Wells & Young, 2004). The following airport terminals can be used (Wells & Young, 2004):

- **Simple unit terminals**: centralised facilities that contain all processing facilities for passengers in one building. Offices and control facilities are also in this building.

- **Combined unit terminals**: one building is shared by more than one airline, but their passenger- and luggage processing facilities are separate.

- **Multiple unit terminals**: each airline has its own separate building (used in larger metropolitan areas) where each building is its own terminal.

- **Linear terminals**: the concept of simple unit terminals, but with extended length to allow more aircraft parking spaces. As the length increases, the walking distances increase, which leads to pier finger terminals.

- **Pier finger terminals**: these are decentralised terminals. Piers/concourses extend from the terminal and aircraft park on both sides of each pier. Some processes are performed in the main terminal while others are performed in the individual concourses.

- **Pier satellite terminals**: these decentralised terminals are similar to the pier finger terminals, but now the aircraft park around a round satellite area at the end of the pier. The advantage of this concept is that satellites can be constructed and expanded without compromising the space between the main terminal and the satellites. This space is necessary for taxi operations.
Remote satellite terminals: satellites are not connected to the terminal, but an automatic people mover is used to transport passengers to the aircraft that are parked at the satellites. This concept was adopted by Atlanta Airport.

In this study, the focus will mainly be on the layout of the airport and not on the functions inside the terminal building. Therefore, the detail involved in these functions is not discussed.

3.2 Airport aprons

An airport apron is the area where aircraft taxi from the runways to the boarding gates. Components of the apron include:

- the aircraft stands (gates)
- the taxiways
- the holding areas
- the holding bays

The aircraft park at the aircraft stands/gates which are connected to the terminal building by loading bridges. Some aircraft stands are designed for large aircraft and some for small aircraft. Small aircraft can also use the large gates, however, that will result in wasted space and the probability of a large aircraft having to wait for a gate to become available will be increased.

The taxiways connect the apron to the runways and allow aircraft to access the runways. Aircraft travel on the taxiways that are parallel to the runways and then enter or exit the runways via taxiways perpendicular to the runways. Aircraft can also pass other aircraft in congested areas via bypass taxiways. Aircraft that have just arrived may not interfere with aircraft that are on the taxiways, ready for take-off. Aircraft must be provided with the shortest possible routes to the runways via the taxiways. Usually, taxiways are situated at many points on the runway to allow aircraft to exit the runway as quick as possible. Some taxiways are designed so that aircraft can exit the runway at high speed. These
3.2 Airport aprons

taxiways connect the runways to the parallel taxiways with a 30 to 45 degree angle instead of being perpendicular to both. Taxiways must be planned in a way that will allow minimum crossing with runways. Other airfield areas include holding areas, where aircraft wait close to the runway for final clearance before takeoff, and holding bays which are situated on the apron and where aircraft can park when no gates are available (Wells & Young, 2004).

The only one of these components that influences the passenger movement and passenger walking distances is the aircraft stands. In this study, the aircraft stands will be referred to as gates. The problem of assigning flights to gates is fundamental for airport efficiency and will be discussed in chapter 10 and chapter 11. This study will specifically focus on reducing the passenger walking distances by using metaheuristic optimisation to find the best flight-to-gate assignment schedule. In Figure 3.1, an example of the walking distance from the aircraft to the gate (and vice versa) is illustrated.

3.2.1 Aircraft parking

Aircraft can park in five different manners:
3.2 Airport aprons

- **Nose-in:** This way of parking requires the least amount of space. The aircraft directly faces the terminal and is connected to it with a loading bridge. Aircraft can enter this parking space on their own, but need a tug to be pushed out of the parking space and to be oriented correctly to move without conflicting with other aircraft. In this way of parking, only the front doors are used for disembarking passengers since the rear door is too far from the terminal to be connected with a loading bridge. This way of parking is shown in Figure 3.2.

![Figure 3.2: Aircraft parking: nose-in](image)

- **Angled nose-in:** The aircraft can now manoeuvre in and out of the parking space on its own while it is brought as close as possible to the terminal. Here, air stairs are used to board and deplane passengers. This way of parking requires more space. This is illustrated in Figure 3.3.

- **Angled nose-out:** The aircraft cannot be brought very close to the terminal, since the propellers may cause damage to the building. This method is mostly used by larger aircraft at airports with little activity and is shown in Figure 3.4.

- **Parallel:** This way requires the largest amount of space. However, it is the easiest way for aircraft manoeuvring. This method is primarily used by smaller aircraft. This way of parking is shown in Figure 3.5.
3.2 Airport aprons

- **Remote:** When all parking spaces next to the terminal/gates are occupied, aircraft can park on a designated parking space away from the terminal (on the apron). Passengers are then transported to the terminal (or from the terminal) by a shuttle or bus. Figure 3.6 illustrates this concept.

Most airports use various types of aircraft parking that suit the different aircraft types and sizes (Wells & Young, 2004). It is difficult to determine the number of gates necessary at an airport for efficient operations. Factors to consider include aircraft sizes and types, the number of aircraft that are scheduled to use a specific gate, as well as the turnaround and gate occupancy time. There must be at least one suitable parking for each type of aircraft at the airport.
3.2 Airport aprons

Figure 3.5: Aircraft parking: parallel

Figure 3.6: Aircraft parking: remote

3.2.2 Gate using

According to Wells & Young (2004), gates can be used on either an exclusive-use, shared-use or preferential-use agreement. These agreements are discussed in this section.

In an exclusive-use system, the air carrier has sole authority over a specific gate. Thus, the gate will be available to the carrier at all times, regardless of changes in the schedule. This system, however, leads to low overall gate use efficiency, since the gate is idle and cannot be used by another aircraft when the carrier is away.

In the shared-use system, gates are shared by more than one carrier and gate use schedules are managed in coordination with other air carriers and airport
3.3 Airport runways

The demand in air transport is continuously growing and the expansion of the physical infrastructure of an airport, such as runways, aprons and terminals, is limited due to a lack of space. These factors lead to a continuous increase in congestion at airports. Solutions that are already in place include the management of airline schedules and other methods to control and reduce congestion and delays. All these solutions lead to more efficient use of airport capacity, and also to an increase in runway capacity. The latter can be accomplished by using innovative operations, by building more runways or both (Janic, 2008).

Airport runway configurations can be different under different weather conditions and traffic volumes. These configurations can be single runways, pairs of parallel runways, pairs of intersecting runways and combinations of these. The operation of parallel runways depends on the weather conditions as well as on their spacing. At many US airports, these parallel runways can operate independently under Visual Meteorological Conditions (VMC), in which visual approaches with traffic in sight are allowed by the Air Traffic Control (ATC). Instrumental Meteorological Conditions are low visibility conditions under which flight rules are no longer visual, but instrumental (IFR). Low visibility conditions occur when visibility is below three nautical miles (nm) or when the ceiling is lower than 1 000 feet. Pilots cannot see each other anymore and the operation of the parallel runways are greatly affected by spacing and other geometry (Janic, 2008).

The runways must be long and wide enough, since aircraft require the minimum allowed distances for take-off and landing. Most large aircraft require between 6 000 and 10 000 feet (between 1 829 m and 3 048 m) for take-off at sea level. Runways are usually between 50 and 200 feet (between 15.24 m and 61
3.4 Concluding remarks on chapter 3

An overview of each component of an airport, the terminals, the apron and the runways, was given in this chapter. Different types of airport terminals were discussed with regard to passenger and luggage facilities, airline control, aircraft gates and operating authority (centralised or decentralised). The functions of the gates, taxiways, holding areas and holding bays on an airport apron were also discussed. Special attention was paid to the gates to which aircraft are assigned and the movement of aircraft on the taxiways. Finally, runway configurations and capacities were explained. In chapter 4 the importance of efficiently managing the capacity of an airport in order to prevent conflicts between aircraft and vehicles will be discussed. These conflicts occur on the apron as well as on the runways.
Chapter 4

Airport capacity and conflicts

The performance of airline industry competitors and the superiority of a specific airline in the industry are the two factors by which the performance of airlines is affected. In order to serve the increasing demand in air travelling in the best possible way, airlines need to compete harder. Airlines have started banding together for the purpose of being able to handle the rapid demand increase. The increase in air traffic has definitely been a burden on airport capacities and this issue must be addressed by efficiently managing the airport capacity in order to
meet the higher demand. This can be achieved in two ways. The first method is to increase the capacity of the airport and the second is to manage the capacity so that it can be utilised more efficiently. Regardless of which method is used, it is inevitable that the time allocated to a flight to land or take off must be efficiently managed and that the environment must be protected (Abeyratne, 2000). In this chapter, airport capacity and conflicts at an airport are discussed.

4.1 Airport capacity

Each of the different components of an Air Traffic Network, including the airports, the airways and the airspace subsets (or sectors), has its own limited capacity. The capacity of an airport, which consists of the number of landings and take-offs per hour, can be determined with high accuracy. The capacity of the airspace subsets depends on the number of aircraft movements that can be controlled simultaneously by controllers of that subset within a certain time interval. The problem is that (during the past few decades) air traffic density has increased substantially which increased the pressure on the air traffic networks. The capacities of the air traffic networks have, however, not increased accordingly (Andreatta et al., 1998).

Peak traffic times at airports are due to passenger preference for travelling at certain times of the day, their preference for travelling certain times of the year, as well as seasonal fares. Although these seasonal fares are considered by some as creating instead of reducing peaks, it is beneficial in an overall view as it in fact spreads the traffic over the year and slows the growth of traffic during the summer season.

Wide aircraft are seen by some airports as contributing to peak traffic especially due to a lack of space in the apron and terminal areas. However, these aircraft cause a delay in runway saturation, leading to the postponement of costly runway expansions (Abeyratne, 2000).

The issue of the increasing number of passengers that stop at the airport between two flights places pressure on the terminal capacity with regard to factors such as luggage handling and the gate-lounge space. This leads to facilitation
problems during peak times. By optimising the aircraft utilisation, similar issues will appear. In some airport capacity management strategies, aircraft are transferred to other airports if the destination airport cannot handle the demand. However, this strategy has been evaluated and it has been concluded that it is counterproductive, since it results in a loss of revenue to the destination airport and that it creates peaking problems at the airports to which the aircraft were transferred. More facilitation problems are created by incompetent government procedures for control at arrival and departure gates and these also increase airport congestion (Abeyratne, 2000).

This peaking problem can be solved by finding ways to accommodate the traffic during peak times and by efficiently managing the traffic flow. The traffic can be accommodated either by the expansion of the airport facilities or by more efficient management of the existing facilities. Another option for handling peaking is to increase the price of flying during these peak times and reducing the off-peak prices. This approach will spread the traffic more evenly over the day, but only in cases where the demand depends on the price. In Germany, the peaking problem has been improved by making school holiday dates different for different areas of the country. This resulted in more evenly spread traffic over the year (Abeyratne, 2000).

4.2 Conflicts at an airport

Due to the increase in air travel demand, there is a general increase in the number of aircraft and land vehicles (buses, operation handling vehicles, service cars and tank trucks) on the airside (apron and runways). In some areas only aircraft are allowed (the runways) while in others only land vehicles (the maintenance areas). However, conflict may arise during passenger transfer to and from aircraft and during the handling and servicing of aircraft where the use of more than one type of vehicle is required. Conflict may also arise in the restricted areas between different aircraft or land vehicles. During operations such as landing and take-off, this is a particular challenge since aircraft speeds are high (Postorino et al., 2006).

A runway incursion is “any occurrence at an airport involving an aircraft, vehicle, person or object on the ground that creates a collision hazard or results
4.2 Conflicts at an airport

in a loss of separation with an aircraft taking-off, intending to take-off, landing or intending to land”, as defined by US Federal Aviation Administration (2002). Runways are the most critical issue regarding conflict, but conflict can also occur with circulation on aprons and taxiways (Postorino et al., 2006).

4.2.1 Conflicts due to runway crossings

Programmes that have been developed to handle the problem of increasing demand in air travel include the Center-TRACON Automation System (CTAS) that is used to improve the efficiency in handling arriving flights at an airport. This programme has proved to be very successful in meeting its objectives, but as the efficiency of flight arrivals improves, the traffic on the airport surface becomes a problem. Surface traffic efficiency can be improved by considering the following three options (Cheng et al., 2001):

- to increase the usable space at the airport, including the runways, taxiways and apron (terminal ramp area/ aircraft parking area)
- to make operational alterations such as changing the runway configuration or reducing the requirements based on the distance by which the different flights should be separated
- to use newer equipment and technology and automation to improve efficiency

However, in order to handle the increased traffic, it may be unavoidable to increase the number of runways and taxiways. When this is done, the complexity in the configuration of the airport will be higher. In most cases, when runways are added, traffic between other runways and the apron will be blocked and more taxiways and runway crossings will be required. Furthermore, if the efficiency of the airport is increased by making operational changes such as decreasing the separation requirements between flights, the traffic density of the arrivals in outer runways will be even higher which will result in more runway crossings. Since the traffic density of the arrivals for inner runways will also be higher, there will be less time for flights on outer runways to cross the inner runways in order to get
to the apron. Thus, by making operational changes to accommodate the increase in demand, efficiency will be lower (Cheng et al., 2001).

In the study done by Cheng et al. (2001), the improvement in traffic efficiency on the surface of the airport by using automation technologies is discussed. Runway crossings is a big issue in the efficiency of airports with a high demand in departing and arriving flights and that have complex runway configurations to handle this demand. The runway crossing requirement becomes complicated when the number of runways and the traffic are increased. Flights sometimes have to line up at the runway crossing point or taxiway while waiting for a gap to cross the runway. These flights then experience considerable delays at the taxiways. By trying to increase the efficiency of arriving flights by reducing separation requirements on the inner runways, a reduction in the time window available for flights on the outer runways to cross will occur. If these crossings can be made without waiting at the taxiways and without causing delays in arriving or departing flights, substantial savings can be made.

Cheng et al. (2001) state that since the aircraft use gas while waiting to cross the runway, and since they may cause departing or arriving flights to be delayed, it will be better for departing flights to be delayed at the gates/parking spaces than at the taxiways. The goal is to minimise the amount of taxi time for departing and arriving flights since this will maximise savings, even though it may result in gate delays.

Because of the complex runway configurations at the Dallas/Fort Worth International Airport (DFW), Cheng et al. (2001) proposed two ideas to prevent queuing at the taxiways. The first idea is a “perimeter taxiway” that will allow the arrivals traffic to approach the apron by going around the ends of the other runways. The problem with the “perimeter taxiway” is that construction will be expensive and that efficiency will be reduced due to the fact that time and fuel spent will be increased since the aircraft will have to travel a longer distance. The other idea is a “rotational runway use” that will group all the arrivals at the one side of the airport and all the departures at the other side. The “rotational runway use” will not eliminate runway crossings, since arriving flights will still have to cross arrival runways. Cheng et al. (2001) envisioned a surface traffic
4.2 Conflicts at an airport

control automation system that will allow better coordination of traffic of arriving and departing flights and will control runway crossings in a tight manner to minimise delays. The success of this system depends on the ability of the aircraft to execute the runway crossing precisely within tight time margins. This is called precision taxi.

An increase in traffic density will result in more runway crossings that have to be accomplished within shorter time windows. To do this in a safe manner, the taxi operations have to be improved. In order to minimise the runway crossing time, the speed by which it is done should be maximised, and to maximise the runway crossing speed, the aircraft should start the crossing at the maximum permissible speed. The minimum runway crossing time will thus be accomplished if the aircraft can cross the runway without having to stop. This will save taxi time, but more importantly, it will reduce the impact on arriving and departing flights that are landing or taking off. This will in turn lead to more runway crossing opportunities. Furthermore, flights lining up at the taxiway will be reduced and fuel efficiency will be increased due to a reduction in braking and accelerating and engine idle time. To accomplish this system successfully, flights need to arrive at the taxiway at the exact instance when it should cross the runway without having to stop and without delaying other flights. A very high level of control is thus required. Therefore, when considering runway crossings, two factors are important, namely the time required to cross the runway and the accuracy of arrival times of the aircraft to cross the runway without having to stop (Cheng et al., 2001).

4.2.2 Conflicts between passengers, vehicles and aircraft

Surface movement ground control systems (SMGCS) have been developed to ensure that the operations at the airport are being performed safely. These systems are used to meet safety requirements and to optimally manage ground movements by controlling the circulation of vehicles on the ground. Ground movements can be optimally managed by directing vehicles along paths that will allow optimal circulation and by reducing excessive aircraft spacing (Postorino et al., 2006).
4.3 Concluding remarks on chapter 4

Pilots and drivers are directed by the guidance function through information about the speed that must be maintained and the path that must be followed. This is done by means of visual aids. The final function, namely the Control function, is used to prevent collisions and runway incursions. This function must thus ensure movements that are safe, quick and efficient. Both pilots and controllers are responsible for this function following “see and avoid” rules. The advanced control function must identify problems and provide solutions for them and it must verify that the required distance between aircraft are kept to ensure safety. Furthermore, it must warn against incursions through an alarm system, it must ensure pilot and driver coordination, it must ensure minimum delay and maximum utilisation of airport capacity through suitable spacing of aircraft and it must separate movements from restricted and secure areas (Postorino et al., 2006).

4.3 Concluding remarks on chapter 4

The demand at airports is continuously increasing and the importance of managing an airport to be able to handle this increase in demand was discussed in this chapter. An airport must either be expanded or managed in a way that will ensure efficient execution of operations. One way to overcome the problem of a demand that is too high, is by providing specials and seasonal fares to spread the traffic at the airport. Due to an increase in demand, more flights arrive at and depart from the airport. This increases the pressure on the runways. The concept of runway crossings was discussed and methods and runway configurations for solving this problem were explained. Furthermore, safety requirements and management in terms of movement of aircraft, surface vehicles and passengers on the apron to avoid collisions were discussed. This chapter concludes the literature review on the background of an airport. In chapter 5 an overview of the problem to be solved will be provided. The different airport designs will be discussed and the problem of assigning flights-to-gates will be outlined.
Chapter 5

Overview of the experiments in the study

In this study the expansion of the Lanseria International Airport is investigated. When future demand in air traffic gets too high at O.R Tambo International Airport, traffic will be directed to Lanseria International Airport. Expansion of Lanseria requires a study of airfield layout to ensure efficient passenger-to-aircraft and passenger-from-aircraft flow. This will be done based on the layout
concept of Hartsfield Jackson Atlanta International Airport. This airport is considered to be the busiest airport in the world. In this chapter the scope of the study is discussed and the problem is explained.

## 5.1 Atlanta International Airport

When referring to Atlanta International Airport, names such as Hartsfield Jackson Atlanta International Airport, Atlanta Airport and Hartsfield-Jackson are commonly used. This airport is located in Atlanta, Georgia, United States and is 11 km south of Atlanta’s central business district. Atlanta Airport serves approximately 88 million passengers per year and is the central hub of the Delta Airlines, AirTran Airways as well as Atlantic Southeast Airlines (Anna Aero, 2011). The airport has 154 domestic gates and 28 international gates and accommodates 970,235 flights each year (City of Atlanta, 2007). The airport’s international services include flights within North America, to and from South America, Central America, Europe, Asia and Africa (Atlanta Airport, 2007).

Atlanta Airport was founded by Mayor Walter Sims in 1925 on an abandoned auto racetrack of 11.6 ha. It was named Candler Field after former Atlanta mayor Asa Candler. The first flight to Candler Field was from Jacksonville, Florida, on September 25 in 1926. Candler Field was a busy airport from the start and was ranked third for number of daily flights of sixteen arrivals and departures. The first and second ranks were New York and Chicago respectively (Franklin, 1954).

The airport was declared a military airfield in 1940 and was used by the Air Force for servicing transient aircraft. Atlanta Army Airfield and Candler Field were now jointly operated. Although Atlanta Army Airfield closed after World War II, the airport doubled in size during this time and set a record of 1,700 arrivals and departures in one day. Candler Field was renamed to Atlanta Municipal Airport in 1946 and in 1957 more than two million passengers were served by the airport making it the busiest airport in the country. Furthermore, between 12:00 and 14:00 each day, Atlanta Municipal Airport became the world’s busiest airport (Atlanta Airport, 2010a).

An expansion in 1961 lead to the airport being able to handle more than six million travellers a year. However, in the same year, the airport had to serve
5.1 Atlanta International Airport

nine and a half million passengers (Henderson, 2008). Further expansion was clearly necessary, and this began in 1967 under the administration of the mayor of Atlanta at that time, Maynard Jackson. The new, expanded airport, named after the former mayor, became the William B. Hartsfield Atlanta International Airport and was opened on 21 September 1980. The expansion project was on-time and under budget as a $500 million project. The airport could now handle 55 million passengers in a year (Atlanta Journal-Constitution, 2003).

Further construction began in May 2001 to build a 2 700 m fifth runway. In 2003, the airport’s name was changed to Hartsfield-Jackson Atlanta International Airport in honour of Maynard Jackson, who was the mayor during the expansions in 1967 (Atlanta Airport, 2010a).

Taxiway Victor, an end-around taxiway, opened in April 2007 and allows aircraft that land on the north runway to access the gate area without delaying or preventing take off of other aircraft. Taxiway Victor was expected to save $26 million to $30 million in fuel (Tharpe, 2007). Take-offs can be continued since the taxiway is dropped approximately 9.1 m from the runway elevation.

Atlanta Airport has a North and a South terminal where passenger check-in and baggage claim take place. Between these two terminals, which form part of a larger building, are a large seating area, concessionaires, a bank, security checkpoints, car rental agencies as well as a MARTA (Metropolitan Atlanta Rapid Transit Authority) train station. Passengers board their flights via six concourse buildings that are parallel to one another. The first of these, known as the T-gates, is connected to the main terminal, with the other five (A, B, C, D and E) spaced parallel to the T-gates.

The sixth concourse, concourse E, is used for boarding of international flights. This concourse was opened in 1994 to handle the demand which resulted due to the Summer Olympic Games which was held in Atlanta in 1996 (Atlanta Airport, 2010a).

Atlanta Airport has an underground transportation mall that starts at the main terminal and connects the six concourses by passing underneath the center of each. The transportation mall also has an automatic people mover that has a starting station at the main terminal as well as a station at each of the other
concourses. The baggage claim area is underneath the main terminal and has its own station (Atlanta Airport, 2010b).

5.2 Different airport designs to be compared

In this study, different airport layouts, including the existing layout of Atlanta airport, are compared via a simulation model of each. The designs are then evaluated, based on passenger movement and convenience, and aircraft movement and traffic, as well as a flight-to-gate assignment programme to optimise passenger walking distances at airports.

Each airport design has a main terminal building where check-in, security checks and passport control are performed. The boarding gates (i.e. aircraft parking spaces) are on opposite sides of five concourses. The concourses are narrow buildings connected to the main terminal building via one or more underground buildings/tunnels. These tunnels connecting the concourses and the terminal building are beneath the surface of the apron in order to avoid conflicts between passengers and aircraft. Each passenger thus moves from the main terminal building to the concourse at which his/her flight is parked via a tunnel. Once the passengers arrive at the correct concourse they move to the aircraft gate where boarding takes place.

The gates, at which the aircraft park, are of different sizes. The concourses are numbered A to E, where A is on the one end of the airport and E is on the other end. In each design, the gates at concourses A and E, looking away from the other concourses are the largest. Aircraft with a wingspan of up to 80 m can park there. The gates at concourses A and E, looking towards the other concourses as well as the gates at concourses B and D looking away from concourse C, are medium sized gates. These gates are for aircraft with a wingspan of up to 65 m. The other gates at concourses B and D as well as all the gates at concourse C are small gates. Aircraft with a wingspan of up to 40 m can park there. All the concourses are of the same length. Nine gates are on either side of the concourses that will accommodate the large and medium aircraft and 18 gates on either side of the concourses that will accommodate the small aircraft. In all the apron designs, each concourse is 55 m wide. The distance between the terminal building
and the concourses is 124.5 m and the distance between any two concourses is 245 m. The gates on either side of each concourse are spaced evenly.

Figure 5.1: Airport apron layout: Design 1

The different airport designs considered in this study are illustrated in Figures 5.1, 5.2, 5.3 and 5.4. **Design 1** is based on the concept of Atlanta Airport, but customised for South African conditions. The five concourses are spaced parallel to the main terminal building as well as to each other. In this case, an underground tunnel runs from the main terminal building to the furthest concourse, connecting all the concourses to each other. At Atlanta Airport there are a lot of transfer passengers due to the fact that Atlanta is a hub for many airlines. These transfer passengers do not need to go back to the main terminal building before boarding the next flight, they only need to move to the concourse of the next flight. Therefore, it is essential that the different concourses be connected to each other. However, if this concept is used in South Africa at the new Lanseria Airport, there will not be as many transfer passengers and it will not be necessary for passengers to be able to move from one concourse to another (Lagus, 2010).
5.2 Different airport designs to be compared

Furthermore, there will not be an automatic people mover at the new Lanseria Airport as is used at Atlanta Airport. The tunnel connecting the concourses will have walking space as well as pedestrian walkways. The runways are perpendicular to the main terminal and the concourses and are on both sides of the airport. The parking space for cars is on the other side of the main terminal building.

Design 2 is the design proposed by Virtual Consulting Engineers in Pretoria, the industry partner in charge of the project of expanding Lanseria International Airport. In this design, the concourses are parallel to each other but perpendicular to the main terminal building. The space between the main terminal and the concourses will be used for aircraft taxiways. Thus, five short tunnels will be built underneath the taxiway, connecting the main terminal to each concourse. Again, the tunnels will have walking space as well as pedestrian walkways.

The runways are perpendicular to the main terminal building and parallel to the concourses, on both sides of the airport. Again, the parking space for cars is on the other side of the main terminal building, opposite the concourses.
5.2 Different airport designs to be compared

In Design 3, the main terminal building will be built from the middle of the first concourse up to the middle of the last concourse, dividing each of the five concourses in half. In this case, concourses are automatically connected to the main terminal building without the need of a tunnel.

In this design, the runways are again parallel to the main terminal building and perpendicular to the concourses. The parking space for cars is at the one end of the terminal building and passengers will all have to enter the terminal from that side.

In Design 4, the third or middle concourse becomes the main terminal building. The parking space for cars is at the one end of the terminal building and because of that, the tunnel connecting the concourses to the main terminal is at the parking space end of the terminal. If the tunnel is positioned in the middle of the concourses as in Design 1, all passengers will need to walk at least half the length of the terminal even if their boarding gates are closest to the parking space end of the terminal. They will then need to walk all the way back to the
5.3 Flight to gate assignment in the study

Figure 5.4: Airport apron layout: Design 4

parking area side of the airport in order to reach their boarding gates. However, if the tunnel is at the parking area side, where all passengers will have to enter the building in any case, passengers will only need to walk the distance to their boarding gates once.

5.3 Flight to gate assignment in the study

In order to compare the different airport apron designs, the simulation models have to assign the different flights to appropriate gates. This is done based on three different rules.

The first model of each design was built using a built-in rule (Rule 1) for assigning flights to gates. This rule works as follows: When a small aircraft arrives, it is assigned to the available small gate nearest to the terminal building. If no small gates are available, the aircraft is assigned to the available medium gate nearest to the terminal building. If no medium gates are available, it is
assigned to the available large gate closest to the terminal building. When a medium aircraft arrives, the same process is followed, except the small gates are not taken into account, since a medium aircraft will be too big for the small gates. When a large aircraft arrives, it is assigned to the available large gate closest to the main terminal building. If no gates of the correct size are available for the arriving aircraft, the aircraft circles above the airport until a gate becomes available.

The second model of each design was also built using a built-in rule (Rule 2) for assigning flights to gates. However, in this rule, a small aircraft can be assigned to a small, medium or large gate without first having to consider all the small gates. Thus, a small aircraft can be assigned to any available gate. A medium aircraft can be assigned to a medium or large gate without evaluating all the medium gates before considering a large gate. Large aircraft can still only be assigned to large gates. Thus, when this rule is used, an aircraft is assigned to the available gate closest to the terminal building if the aircraft is not too big for that gate. This rule is tested to determine whether the fact that small aircraft can now occupy the medium and large gates results in medium and large aircraft having to wait to be assigned to gates.

The third model of each airport design was built using metaheuristic optisation to optimise the flight-to-gate assignment process. In these models, the cross-entropy method was used to find the flight-to-gate assignment schedule that will reduce the passenger walking distance. This will be explained in Chapter 11.

5.4 Considerations in the study

The following considerations are made in the study:

- The movement of aircraft on the runways is outside the scope of the study. The airports are only compared based on the layout of the terminal building and the concourses. Also, the flight schedule is based on that of O.R Tambo International Airport and slot coordination methods for arriving and departing flights are not considered. The only time spent and distance
5.4 Considerations in the study

travelled by aircraft that are taken into account are that on the taxiways and not on the runways.

- Aircraft are parked nose-in at the gates. This is because an aircraft will not necessarily leave the gate to taxi in the same direction as when it arrived. It may happen that the aircraft will taxi in the other direction. Therefore it must not be parked in a way by which it will be forced to leave the gate with an angle wider than 90 degrees.

- The side from which an aircraft will enter the runway depends on the direction of the wind. Since an aircraft has to descend and ascend against the wind, the aircraft will have to enter the runway from the correct side.

- The taxiways used by the aircraft to get from the gates to the runway and from the runway to the gates are those that will result in the shortest taxiing distance for the aircraft.

- The only delays considered at the airport are those caused by the airport itself, thus delays due to passengers not reaching the gate before the departure time of the flight or due to an aircraft having to wait for a suitable gate to become available. No delays outside the control of the airport are considered.

- Every gate can be used by any airline.

- The simulation is performed with only one replication since there are no random numbers in the model. However, even though there are no random numbers, simulation is still a suitable technique because of the complexity of this problem. Also, delays due to passengers not reaching the gates in time (due to the passenger walking speeds), or the time of an aircraft waiting at the end of a taxiway for another aircraft to exit that taxiway in order to prevent a collision, can be modelled realistically by using simulation. Furthermore, the model is dynamic as it evolves over time and simulation is therefore an appropriate technique.
5.5 Concluding remarks on chapter 5

The simulation and metaheuristic optimisation models were built while considering the above mentioned factors. These models are discussed in chapter 7 and chapter 11 respectively.

5.5 Concluding remarks on chapter 5

In this chapter the study was outlined. The airport concept used in the design of the different apron layouts was discussed and the different scenarios to be simulated and compared were explained. The objective, constraints and rules to be used in the flight-to-gate assignment procedure in the study were discussed. The objective of the flight-to-gate assignment process is to minimise the passenger walking distances at the airport. The three rules that will be used to assign flights to gates were described. Furthermore, the scope of the study was outlined by pointing out the considerations made. The problem solving phase of the study will be carried out in the following six chapters. The first part of the problem solving phase is the simulation study. This part is started in chapter 6 in which an overview of simulation will be provided. The overview will be given in preparation for the discussion of the simulation models developed for the purpose of the study.
Chapter 6
Simulation as a problem solving technique

Throughout literature, simulation is a widely used technique. As defined by Kelton et al. (2004), simulation is the imitation of a real world process or system and its behaviour. By using simulation, the process or system’s performance can be measured, its operations can be improved and a new process or system can be designed. Furthermore, it can be used to get a good understanding of how the system operates.
6.1 The application of simulation

According to White & Ingalls (2009) a model is an entity that represents another entity when it is impractical to investigate the actual system. Investigation of the actual system may be too expensive, unsafe, slow or disruptive. Models can also be used to study conceptual systems. Via simulation, system behaviours are imitated and systems are experimented with to create observations from each behaviour. Simulation can be used in trying to understand and analyse a system, to test and compare alternative designs, to validate a design, to explain outcomes and to investigate recommendations.

In this chapter, an overview of a simulation study is given with specific regard to the application of simulation, the advantages and disadvantages of simulation, the components of a simulation model and the steps in a simulation study.

6.1 The application of simulation

The application of simulation falls into two categories. The first category is simulation for training or entertainment. Many professionals develop skills in a simulated environment that is protected from failure due to inexperience. This training concept is illustrated in a flight simulator in which pilots learn to operate an aircraft. Computer games, where one drives a car or something similar also fall into this simulation category. In the second simulation category, processes are analysed and designed. It is in this category that engineers and operations researchers mostly perform their simulations. Continuous system simulation, discrete-event simulations, agent-based simulation, Monte Carlo simulation and hybrid simulation are all categories of simulation that have different implementation strategies (White & Ingalls, 2009).

For the purpose of this study, discrete event simulation will be used. When the state of the simulation model changes at discrete points in time which can possibly be random, the simulation is a discrete-event simulation.

In the “transaction-flow world view”, a system is seen as containing traffic units that are discrete and that compete for the use of a scarce resource while moving through the system. This view usually provides the foundation for discrete-event simulation. A unit of traffic is called an entity (Schriber & Brunner, 2009).
6.2 Advantages and disadvantages of simulation

The characteristics, behaviour and relationships of the system are described by the system theories. By experimenting with the system, system data and results are collected. By abstracting system observations and hypothesising the data and results, system theories are obtained (Sargent, 2009).

A model of a real world system is created by firstly developing a conceptual model of the system. The conceptual model is a representation of the real world system and is developed by modelling the system theories. The specifications of the model are a description of how the conceptual model is implemented on a computer system. After implementing the conceptual model on an adequate computer system, the model is called the simulation model. This model is run on the computer and it is experimented with. The data and results obtained from these experiments are the simulation model data and results (Sargent, 2009).

In defining the objectives of the simulation study, the analyst must know the stakeholders as well as their interests in the project and what they need from the study for it to be successful. A stakeholder is anyone who is interested in or affected by the project. Different stakeholders are involved in each project, however, some are more important than others. After knowing how the stakeholders define and measure the success of the project, the analyst can determine its objectives (Sturrock, 2009).

6.2 Advantages and disadvantages of simulation

The following advantages of simulation are discussed by Banks (1998):

- Through simulation, changes to a system can be tested and experimented with, without having to use any resources.

- In a simulation study, time can be expanded or compressed.

- Through simulation, certain answers about real world systems can be answered.

- New policies or methods can be explored, without disrupting the real system.
6.3 Components of a simulation model

- Problems can be diagnosed and insight can be gained in a complex system.
- Through simulation, constraints can be identified.
- An understanding can be gained on how a system really operates.
- A plan and the running of a facility can be visualised.

Banks (1998) further identifies the following disadvantages:

- The development of a model that imitates a real-world system is a complex art.
- The output of a simulation model may be random variables and may be difficult to interpret.
- Inappropriate use of simulation may occur.

Despite these disadvantages, simulation is an appropriate technique for the experiments in this study, as stated in sections 2.1 and 5.4. By using simulation, future airport concepts can be tested and the dynamic nature of airports can be modelled.

6.3 Components of a simulation model

Different components present in a simulation include the inputs, outputs and states, the entities and their attributes, activities and events, resources and statistical collectors. These components will be discussed in section 6.3.1 to 6.3.5.

6.3.1 Inputs, outputs and states

The actions of the environment that influence the system are the inputs to the system. In an airport traffic problem one input is the arrival rate of the aircraft. The state of the system depends on changes in the system condition that are caused by the inputs. The state of the airport would be the number of aircraft on ground at any given time, in other words, the level of congestion at the airport. The outputs of the system are needed to answer questions about the system and
6.3 Components of a simulation model

can be derived from the system state. In the airport problem, one of the outputs can be the average time spent by an aircraft at the airport (White & Ingalls, 2009).

6.3.2 Entities and attributes

In discrete event simulation, the arrival of entities is the input for the simulation. The entities flow through the system and change the state of the system variables. In the airport problem, the aircraft and the passengers are the entities. These entities have unique characteristics, called attributes. The attributes of an aircraft can be its size and its arrival time (White & Ingalls, 2009).

Entities can be external and explicitly created and manipulated by the modeller, or internal and implicitly created and manipulated by the software. A resource provides a service to entities and often has limited capacity. Entities then have to compete or wait to be served by the resource.

When an entity has to wait for the use of the resource, it experiences a delay. As entities move through the system, they change states. These states are the active state, ready state, time-delayed state, condition delayed state and dormant state. An entity is in the active state if it is moving. It will continue moving until it reaches a delay of some type. It will then go into another state and another entity will possibly enter the active state. In the case where two or more entities need manipulation at the same time, only one will continue moving while the others will experience a delay. The entities that are delayed are known to be in the ready state. These entities are waiting to go into the active state. Sometimes, entities are delayed for a known time period before entering the ready state. These entities are in the time-delayed state. Entities may also be delayed until some sort of condition is satisfied. These entities are in the condition-delayed state. When entities are delayed and cannot move into another state through changes in the condition of the model, they are in the dormant state (Schriber & Brunner, 2009).
6.3 Components of a simulation model

6.3.3 Activities and events

Processes in a simulation are called activities and changes in the state of the system are called events. Entities interact with activities to create events. Activities are divided into three categories: Delays, queues and logic.

A delay is caused when the flow of an entity is put on hold for a definite time period. A delay in the flow of an aircraft would be the time spent by the aircraft at the gate/parking space. This time period can be either constant or random. When the delay of any entity starts, an event occurs. If the time period of the delay is $d$ time units, then the entity will be delayed until the current time plus $d$ time units has expired, after which it will carry on with its movement (exit the parking area to depart) and create another event.

A queue occurs when an entity’s movement is put on hold for an unspecified time period. For example, if all gates are occupied, arriving aircraft will have to wait in some sort of a queue until a gate becomes available.

Logic activities are manipulation of the state variables. Thus, only allowing an entity to affect the state of the system if it is desirable. An example of a logic activity is the decision of whether or not an aircraft is allowed to land and thus affect the level of congestion at the airport (White & Ingalls, 2009).

6.3.4 Resources

Resources in a simulation are those things that have constrained capacities. In the airport problem, the resources are the gates to which aircraft must be assigned and where the aircraft park and wait for the passengers to board (White & Ingalls, 2009).

6.3.5 Statistical collectors

The parts of the simulation used to gather statistics on the values of performance measures or global variables are called statistical collectors. There are three types of statistics, namely counts, time-persistent statistics and tallies.

Counts are used to count for instance the number of aircraft that arrived at the airport in one hour. Time-persistent collectors give a time-weighted average
of a variable, for example the average number of aircraft present at the airport over a 24 hour period. Tally statistics are concerned with only one observation regardless of other observations or the time between them. An example of a tally statistic is the amount of time spent by a specific aircraft at the airport (White & Ingalls, 2009).

### 6.4 Steps in a simulation study

The following seven steps are used in the process of model development (Banks, 1998):

- problem formulation
- objective setting and overall project plan
- model conceptualisation
- data collection
- translation of concept model to computer model
- verification and validation
- experimental design and analysis

These steps were followed in the development of the simulation models in this study. The problem was formulated and the objectives were set in chapters 1 and 5. The development of the concept model and the translation of the concept model to the computer model will be discussed in chapter 7. The verification and validation processes will be presented in chapter 8. The data used in the models and the experimental design and analysis will be discussed in chapter 12.

### 6.5 Concluding remarks on chapter 6

In this chapter, an overview of simulation was given. Also, the different applications of simulation were explained and specific attention was paid to simulation
6.5 Concluding remarks on chapter 6

being used to analyse and design processes. The components of a simulation study such as states, entities, attributes, activities, events, resources and statistical collectors were discussed. Finally, the steps to follow for a successful simulation study were listed. The simulation study will be continued in chapter 7 by discussing, in detail, the simulation models of the different airport designs. The logic in the models will be explained and the first two rules for assigning flights to gates will be discussed.
Chapter 7

The simulation models

In this chapter, the logic in the simulation models is discussed and a concept model of the simulation processes is presented. The different types of entities in the model, the movement of the aircraft and passengers and the loading and unloading of passengers in the model are explained.

The Simio® simulation software was used for model implementation. The reasons for using Simio are because free, unlimited research is allowed, the software was donated to the Stellenbosch Industrial Engineering department and it is the newest, most sophisticated simulation software available.
7.1 The entities in the model

In this chapter, “counter” will refer to a placeholder in the simulation. “Node” will refer to a position on the apron where taxiways split and aircraft have to decide in which direction to travel.

7.1 The entities in the model

The different entities in the simulation model are the aircraft, the arriving passengers and the departing passengers. The process at the airport is as follows:

The aircraft arrive with passengers, then taxi to their assigned gates where they are connected to the terminal building via loading bridges. The arriving passengers disembark the aircraft and walk over the loading bridges to reach the concourses at which the aircraft are parked. The arriving passengers then move to the main terminal building to collect their luggage. Each aircraft stays parked at the gates until it has to leave for the next flight. When the boarding time for the next flight of the aircraft is almost reached, the departing passengers start walking from the main terminal building to the gate from which the flight will depart. They then board the aircraft, after which the aircraft is ready to start taxiing away from the gate to reach the runway.

7.1.1 The aircraft

The flight schedule of the airport is read from a pre-defined data table in Simio. When the arrival time of each flight is reached, according to the schedule, a flight is created. Each flight has the following properties that are assigned to it:

- flight number
- flight type
- number of passengers
- arrival time
- departure time
- boarding time for departing passengers
7.1 The entities in the model

In the simulation model, the flights are numbered from one to the number of flights in the run, with the first flight being the one with the earliest arrival time. The flight type is dependent on the size of the aircraft. The different sizes are:

- type A, where aircraft have a wingspan of 15 m
- type B, where aircraft have a wingspan of 29 m
- type C, where aircraft have a wingspan of 36 m
- type D, where aircraft have a wingspan of 52 m
- type E, where aircraft have a wingspan of 65 m
- type F, where aircraft have a wingspan of 80 m

In the model, the gates are numbered from one to 122. No gate can have the same number as another since it will then not be clear to which gate to direct the aircraft. However, the gates on either sides of each of the taxiways are numbered from one to 18 for those concourses with large gates and from one to 36 for those with smaller gates. This numbering of gates is important for knowing the position of each aircraft on the taxiway in order to prevent a collision of the aircraft. This concept is further explained in section 7.2.1. It is thus necessary that each gate has a property of its position on the taxiway. It is also important to know on which taxiway each gate is in order to know where each aircraft will leave the taxiways perpendicular to the concourses to enter the taxiway at which its gate is situated.

The model runs through all the gates and assigns the position and the taxiway of each gate to a one dimensional matrix. This is described by Algorithm 1.

Algorithm 1 Assign gate positions and taxiways

For $i = 1$ to number of gates

- position of gate($i$) = Gate position (as read from table in Simio)
- taxiway of gate($i$) = Taxiway of gate (as read from table in Simio)

$i ← i + 1.$

end
Once this is done, there is no need for the process to be repeated. By including the condition that it must be the first created flight in order to enter the loop described above, this step is only performed in the beginning of the simulation run.

After considering the above mentioned condition, each aircraft is assigned a priority. The first aircraft has a priority of one, the second of two and so on. The variable `priority` is assigned to the attribute `aircraft.priority` and then the variable `priority` is incremented. When the next flight arrives it will thus have a priority of one more than the previous. These priorities are then used as flight numbers. The flight type, representing the wingspan of the aircraft, is assigned to the aircraft in order to know which gates are suitable for it. This flight type is read from the table containing the flight schedule. As stated previously, in the new designs of Lanseria International Airport, there are three different gate sizes, small, medium and large. When a flight is created it has to be assigned to a gate. In this chapter the first and second rules of assigning flights to gates will be explained. The third rule will be explained together with the metaheuristic in chapter 11.

The first rule assigns the flight to the appropriate gate nearest to the main terminal building since all passengers have to go to that building to collect their luggage. In making this decision, the model starts at the gate closest to the terminal building. If the gate is currently occupied by another aircraft, the next gate is chosen. Once an available gate has been found, the model evaluates it to determine whether it is an appropriate gate for the flight in question. The gate is appropriate if it fits the size of the aircraft. If not, the next gate is evaluated. The logic used in the model is described below:

- If the selected gate is a small gate, and the aircraft is small, the aircraft can be assigned to the gate.
- If the selected gate is a small gate and the aircraft is medium or large, the next gate is considered.
- If the selected gate is a medium gate, and the aircraft is small, the next gate is considered. This is done even though a small aircraft can park at a medium gate, because it may happen that a medium aircraft arrives at the
7.1 The entities in the model

airport with no suitable gates available, while small aircraft are occupying the medium gates. A small aircraft can be assigned to a medium or large gate only when there are no small gates available.

- When a medium gate is selected and it is a medium aircraft, the aircraft is assigned to the gate.

- If the selected gate is medium and the aircraft is large, the next gate is considered.

- If a large gate is selected, small or medium aircraft can only be assigned to it if there are no small or medium gates available. Again, this is done to prevent the situation of having no suitable gates available for the large aircraft.

- If a large gate is selected and the aircraft is large, the aircraft is assigned to the gate.

- If there are no suitable gates available for a specific flight, the aircraft has to wait. As soon as the first gate becomes available, it is tested to determine whether it is suitable for the aircraft in question. If it is, the aircraft is assigned to it. If not, the aircraft has to wait until a suitable gate becomes available.

In the second rule, a small aircraft is assigned to a gate of any size. Thus, if the available gate closest to the terminal building is a medium or large gate, the aircraft is assigned to it, even though there may be small gates available. Similarly, if a medium aircraft arrives and the closest available gate is a large gate, the aircraft is assigned to it. In this rule, it may happen that a medium or large aircraft arrives without there being a suitable gate available while small aircraft are occupying the medium and large gates.

Once a flight is assigned to a gate, attributes such as the taxiway at which the gate of the aircraft is and the position of the gate on the taxiway are assigned to the aircraft. The condition of whether the arrival time of the aircraft has expired (i.e. whether the aircraft had to wait to be assigned to a gate) is tested. If the aircraft did have to wait, the waiting time is recorded. The total waiting time
of all the flights will later be used as an output statistic to evaluate the airport layout.

### 7.1.2 The arriving passengers

The number of passengers property of the flight is used to determine the number of arriving passengers to create for that flight. All the arriving as well departing passengers on each flight are presented by one passenger with a property of the number of passengers on that flight. This can be done since in real life, all the arriving passengers arrive at the same time. These passengers are also created as soon as the arrival time for the flight in question has been reached. The passengers are assigned a priority in the same way as the aircraft. The value of this priority is the number of the flight on which they are arriving.

After the correct number of arriving passengers is created, they are combined with the aircraft. The condition for combining the aircraft and the passengers is the priority/flight number, which must be the same for both. Together with the aircraft the passengers are directed to the gate to which the flight was assigned. The time at which the passengers are created is recorded. This will later be deducted from the time they leave the airport (the time the entities are destroyed) in order to use as an output statistic to evaluate the total time as well as the average time the passengers spent in the system. If the aircraft has to wait to be assigned to a gate, the arriving passengers will automatically have to wait, since they have to be combined with the aircraft before they can move to their gate. This waiting time will thus also be reflected in the time the arriving passengers spent in the system.

### 7.1.3 The departing passengers

When the boarding time for departing passengers property of each flight has been reached, the passengers that need to depart on that flight are created. Since, for the purpose of this study, the departing passengers are only modelled from boarding time and not from check-in, it is assumed that everyone that checked in is on time for boarding and all the departing passengers, as in the case with the arriving passengers, can be represented by one passenger with a property of the
number of passengers on that flight. Again, the passengers are assigned a priority which is the value of their flight number. Some aircraft will not have a flight for which to depart immediately after the arriving passengers disembarked the plane. Sometimes aircraft stay at the airport for hours or even days before having to depart for their next flights. Thus the aircraft will not necessarily depart in the same order as in which they arrived. For this reason, the departing passengers cannot be created in the order of their assigned flight numbers (priorities), since, for example, the boarding time of the tenth flight may be reached before the boarding time of the first flight. The flights thus need to be sorted according to the boarding times in order to create all the departing passengers in the correct order and at exactly the right moment.

In some cases, flights have very little turnaround time. If the airport is too busy and there are no suitable gates available when the flight arrives, it may happen that the aircraft is still waiting to be assigned to a gate after the boarding time for the departing passengers of the next flight has been reached. These passengers then have to wait to know to which concourse and which gate to move. The condition of whether the gate is already known must thus be satisfied before the departing passengers can start walking. If this condition is satisfied, the passengers head for the boarding gates.

The departing passengers know to which gate to go, since the variable \texttt{GateNo(Aircraft.Priority)} or \texttt{GateNo(DepartingPassengers.Priority)} is used, where priority is the flight number as assigned to each aircraft and all arriving and departing passengers. Thus, if the aircraft’s priority is 12, for example, the departing passengers that will leave on that flight also has a priority of 12. And they all have to go to the gate with the number equal to the value assigned to the variable \texttt{GateNo(12)}.

The time at which the departing passengers are created is recorded in order to be able to later calculate the time they spent in the system. Again, this is done to be used as an output statistic, in order to evaluate the airport layout.
7.2 The taxiways in the model

The paths/taxiways available for the aircraft to travel on are as depicted in Figure 7.1.

As shown in Figure 7.1, all the taxiways can be used in both directions. Aircraft cannot pass each other between two concourses except for where there are two taxiways between the concourses. There are two taxiways between concourses B and C and between concourses C and D. There are also two taxiways on either side, perpendicular to the concourses. The process must thus be modelled so that no two aircraft will have to pass each other on any of the taxiways. The taxiways can either be entered from the side, or from the gates at the concourses.
7.2 The taxiways in the model

7.2.1 Entering a taxiway from a gate

In the model, the gates on either side of a taxiway are numbered as shown in Figure 7.2:

![Figure 7.2: Taxiway with gates](image)

The model uses the gate numbers to determine the position of each aircraft on the taxiway. In this chapter, the point at which an aircraft is on a taxiway, defined in terms of the number of the gate situated at that point next to the taxiway, will be referred to as the position of the aircraft. Thus if aircraft A is parked at gate 9 then the position of aircraft A is nine. The place of the aircraft in the line of aircraft travelling in the same direction on a taxiway (i.e. first in line, second, etc.) will be referred to as the place in the line or as the place of the aircraft.

If an aircraft, aircraft A, was parked at gate 9 and now wants to enter the taxiway to go in the direction of gate 1, but another aircraft (aircraft B) is on the taxiway at the point where gates 7 and 8 are and is taxiing in the direction of gate 17, then the aircraft at gate 9 (aircraft A) has to wait until aircraft B has passed gate 9 before entering the taxiway. Whenever an aircraft wants to enter a taxiway, it gets a priority. These priorities are in the order of which the aircraft arrived at the taxiway and the order in which they must enter the taxiway. Thus, if there is a third aircraft (aircraft C) that also wants to taxi in the direction of gate 17, it cannot enter the taxiway simply because aircraft B is taxiing in the desired direction, because aircraft A arrived at the taxiway before aircraft C and must thus be allowed to enter it without having to wait for aircraft C. Only after
### 7.2 The taxiways in the model

Aircraft A has entered the taxiway and all possibilities for a collision have been evaluated and cleared, can aircraft C enter the taxiway.

The model evaluates all collision possibilities as follows:

- If an aircraft (aircraft A) is about to enter a taxiway and wants to travel in the direction of gate 1 and there are no aircraft taxiing in the opposite direction, aircraft A is allowed on the taxiway. This scenario is depicted in Figure 7.3.

![Figure 7.3: Entering taxiway from a gate, scenario 1](image)

- If there are aircraft taxiing in the opposite direction, but the last one in the row has already passed the gate at which aircraft A is waiting, then aircraft A can enter the taxiway since it will already be behind the aircraft travelling in the other direction. Figure 7.4 describes this scenario.

![Figure 7.4: Entering taxiway from a gate, scenario 2](image)
7.2 The taxiways in the model

- If all the flights in the row of aircraft taxiing towards gate 17 have not passed the gate at which aircraft A is waiting, then the model has to evaluate every flight travelling in that direction. If any of these aircraft, for instance aircraft B, has not passed the position of aircraft A, and aircraft B is departing (in other words headed for the end of the taxiway to get to the runways as opposed to heading for a gate), then aircraft B will definitely not get off the taxiway without passing this position. In this example, aircraft B is taxiing in the direction of gate 17. The reason for this is because it will result in the shortest route to the runway from which the aircraft will have to depart. Thus, aircraft A has to wait for it to pass. This scenario is demonstrated in Figure 7.5.

![Figure 7.5: Entering taxiway from a gate, scenario 3](image)

- If there are no aircraft like aircraft B, thus no departing aircraft that have to pass the position of aircraft A, but there is an aircraft (aircraft C) that has not passed this position and aircraft C is an arriving flight (thus headed for a gate), but the gate assigned to aircraft C is on the other side of the gate at which aircraft A is waiting (which means aircraft C will have to pass the gate of aircraft A) then aircraft A is not allowed to enter the taxiway. This scenario is illustrated in Figure 7.6.

- If the gate of aircraft C is not on the other side of the current position of aircraft A (thus the gate assigned to aircraft C is on the side of gate 1), aircraft A can only enter the taxiway right away if the position of aircraft C is closer to the gate assigned to aircraft C than the position of aircraft A is.
7.2 The taxiways in the model

Figure 7.6: Entering taxiway from a gate, scenario 4

to the gate of aircraft C. In this case aircraft C will reach its gate and exit the taxiway before aircraft A gets there. This can be described as follows: if (the gate assigned to aircraft C - the current position of aircraft C) < (the gate assigned to aircraft A - the gate assigned to aircraft C), then aircraft C will reach its gate before the two aircraft meet each other. This scenario is shown in Figure 7.7.

Figure 7.7: Entering taxiway from a gate, scenario 5

If, for example, aircraft A (that wants to taxi in the direction of gate 1) is parked at gate 9, then, as described above, it can only enter the taxiway if there are no aircraft taxiing in the direction of gate 17 that have a current position of less than nine and that want to pass gate 9. However, this results in the problem that if there is an aircraft travelling in the direction of gate 17 with a current position of ten, it seems as though it has passed aircraft A which has a current position of nine while positions nine and ten are in fact at the exact same spot on the taxiway. This problem is resolved by adding the modulus of the value of
7.2 The taxiways in the model

the position divided by two to the value of the position. Thus, if aircraft A has a position of nine, the actual position will be:

\[ 9 + 9 \mod 2 = 9 + 1 = 10 \]  \hspace{1cm} (7.1)

and the position of the aircraft at position ten will be:

\[ 10 + 10 \mod 2 = 10 + 0 = 10 \], \hspace{1cm} (7.2)

which means the position of aircraft A will no longer be smaller than the position of the aircraft at gate 10, and aircraft A will not be allowed on the taxiway until the other aircraft is out of the way.

When an aircraft enters a taxiway, it is important to define its place in the line of aircraft on that taxiway. For example, if there are three aircraft taxiing in the direction of gate 1 and an aircraft enters the taxiway between the first and the second aircraft, to taxi in the same direction, the aircraft that was previously the second one on the taxiway now becomes the third one, and the one that was previously the third aircraft now becomes the fourth aircraft on that taxiway, travelling in the direction of gate 1. The logic used by the model to accomplish this is described below:

- If, for example, aircraft A is entering the taxiway between aircraft B and aircraft C, and behind aircraft C there is another aircraft, aircraft D, taxiing in the same direction. In this example, all of these aircraft are travelling in the direction of gate 1. Thus, at the moment, aircraft B is at the front of the line of aircraft, aircraft C is second in line and aircraft D is third. This situation is illustrated in Figure 7.8.

- A counter is used and the value of the counter is set to one. The condition is tested whether the position, in terms of the gate number, of aircraft A is smaller than the position of the flight at the place in the line equal to the value of the counter (thus the position of the first flight in the line if the value of the counter is one, i.e. aircraft B). In this example, since aircraft A will enter the taxiway between aircraft B and aircraft C, the position of aircraft A is larger than that of aircraft B and smaller than that of aircraft
7.2 The taxiways in the model

The taxiways in the model

C. The condition is thus false when the value of the counter is equal to the position of the first flight (Aircraft B).

- The counter is incremented and the second flight in the line is evaluated. The condition is tested again: if the position of aircraft A is smaller than that of the second aircraft in line (aircraft C), like it is in this example, the condition is true.

- Another counter (counter 2) is now used and the value of this counter is set equal to the number of aircraft on the taxiway, excluding the one entering the taxiway (aircraft A). While the value of this counter is larger or equal to the value of the first counter, the following action is performed: all the properties of the flight on the taxiway with the place in the line equal to the value of counter 2, including the flight number, are assigned to the flight at the place in the line equal to one more than the value of counter 2. In other words, aircraft D, that was the third aircraft on the taxiway travelling in the same direction before aircraft A entered the taxiway, now becomes the fourth aircraft on the taxiway. The value of the second counter is decremented by one and the action is repeated, again for the flight with the place in the line equal to the value of counter 2. Thus, the flight that was previously second on the taxiway (aircraft C) now becomes the third one in the line. This is done until the value of counter 2 is equal to that of the first counter (representing the place in the line of the new aircraft on the taxiway).
7.2 The taxiways in the model

- The properties of the new aircraft are now assigned to the aircraft with the place in the line equal to the value of the first counter. In other words, aircraft A now becomes the second aircraft on the taxiway travelling in the direction of gate 1.

The logic described above can be coded as in Algorithm 2:

**Algorithm 2**: Logic for an aircraft entering a taxiway

1. \( \text{counter } 1 = 1 \).

2. While aircraft A current position > position of aircraft (\( \text{counter } 1 \))
   - \( \text{counter } 1 \leftarrow \text{counter } 1 + 1 \)
   end

3. \( \text{counter } 2 = \text{number of aircraft on taxiway} \)
   While \( \text{counter } 2 \geq \text{counter } 1 \),
   - place in line (\( \text{counter } 2 + 1 \)) \leftarrow \text{place in line (counter } 2 \)
   - position of aircraft (\( \text{counter } 2 + 1 \)) \leftarrow \text{position of aircraft (counter } 2 \)
   - gate of aircraft (\( \text{counter } 2 + 1 \)) \leftarrow \text{gate of aircraft (counter } 2 \)
   - \( \text{counter } 2 \leftarrow \text{counter } 2 - 1 \)
   end

4. place in line (\( \text{counter } 1 \)) \leftarrow \text{aircraft A priority}

5. position of aircraft (\( \text{counter } 1 \)) \leftarrow \text{aircraft A current node}

6. gate of aircraft (\( \text{counter } 1 \)) \leftarrow \text{aircraft A gate number}

Once an aircraft enters a taxiway from a gate, the gate at which it was parked becomes available for other flights. Another aircraft, aircraft E, travelling in the direction of gate 1 can enter the taxiway between any of the aircraft in Figure 7.8 in the same way as described above.

### 7.2.2 Entering a taxiway from the side

When an aircraft (aircraft A) wants to enter a taxiway from the side, it again has to wait its turn according to the priority assigned to it. Assume aircraft A wants to enter the taxiway from the left side:
7.2 The taxiways in the model

- If there are no aircraft coming towards aircraft A, it can enter the taxiway as shown in Figure 7.9.

![Figure 7.9: Entering taxiway from the side, scenario 1](image)

- Aircraft A cannot enter the taxiway if there are aircraft coming towards it, and any of them are departing which means they will definitely not exit the taxiway to park. In this case aircraft A has to wait until those aircraft have passed. See Figure 7.10.

![Figure 7.10: Entering taxiway from the side, scenario 2](image)

- Aircraft A can enter the taxiway if none of the approaching aircraft are departing, and the value of the gate assigned to the aircraft (aircraft B) headed for the gate closest to the left end of the taxiway (from which aircraft A wants to enter) is bigger than the value of the gate assigned to aircraft A (after taking into account the modulus of the gate number divided by two). This is shown in Figure 7.11.
7.2 The taxiways in the model

Figure 7.11: Entering taxiway from the side, scenario 3

- If any of the gates assigned to the aircraft are smaller than that of aircraft A (thus including the gate of aircraft B) then the model evaluates every aircraft coming towards the left side of the taxiway. If the gate of the flight that is being evaluated is bigger than that of aircraft A, it is cleared and the next aircraft is taken into account. If aircraft C has a gate smaller than that of aircraft A, but the distance from the current position of aircraft C to the gate of aircraft C is smaller than the distance from the position of aircraft A to the gate of aircraft C, then aircraft A can enter the taxiway since aircraft C will reach the gate assigned to it and leave the taxiway before aircraft A gets to it. If this is established, the next approaching aircraft is evaluated. This is shown in Figure 7.12.

Figure 7.12: Entering taxiway from the side, scenario 4

- If the distance between aircraft C and the gate assigned to it is not smaller than the distance between aircraft A and the gate of aircraft C, but the distance between the gate assigned to aircraft A and the current position...
of aircraft A is smaller than the distance between aircraft C and the gate assigned to aircraft A, then aircraft A can also enter the taxiway, because it will leave the taxiway before reaching aircraft C. If any of the above mentioned conditions are violated, aircraft A has to wait before entering the taxiway. Figure 7.13 shows how this process is performed.

When an aircraft enters a taxiway from the side, it automatically becomes the last aircraft in line of all the aircraft on that taxiway travelling in the same direction. Entering the taxiways perpendicular to the concourses and entering the taxiways between the runways and the concourses from the middle is done in a similar way to the above described processes.

### 7.2.3 Exiting a taxiway

When an aircraft exits a taxiway it is evaluated to determine whether the aircraft was the last aircraft on that taxiway travelling in a certain direction. If it was, the second last aircraft now becomes the last one. It is further determined whether the aircraft was the one with the smallest gate number (for aircraft travelling in the left direction) or the one with the biggest gate number (for aircraft travelling in the right direction) in order to know which flights are now the ones with the smallest or biggest gate numbers.

When an aircraft leaves a taxiway, the place in the line of the other aircraft travelling in the same direction on that taxiway must be updated. This process is similar to the one described in section 7.2.1, where an aircraft enters a taxiway.
7.2 The taxiways in the model

If, for example, there are three aircraft on a taxiway, aircraft A, aircraft B and aircraft C, all taxiing in the direction of gate 1 and the one in the middle, aircraft B, leaves the taxiway to enter a gate, then aircraft C which was third on the taxiway becomes the second one in line. This logic is described below:

- If it is the second aircraft in the line that leaves the taxiway, then a counter \((\text{counter } 1)\) is assigned the value two. Thus the value of the counter equals the value of the place in the line of the exiting aircraft. This situation is illustrated in Figure 7.14.

![Figure 7.14: Exiting a taxiway](Image)

- While the value of this counter is smaller than or equal to the number of aircraft on that taxiway, including the aircraft leaving the taxiway, the following process is executed: all the properties of the flight that was at the place in the line with the value of \(\text{counter } 1\) plus one, are assigned to the flight at the place in the line with the value equal to that of \(\text{counter } 1\).

- The value of \(\text{counter } 1\) is incremented and the process is repeated until this value is bigger than the number of aircraft on the taxiway.

- Since there are now only two aircraft on the taxiway, the position of the third aircraft in the line, which is not there anymore since that aircraft moved to the second place in line, must be big enough so that the position of an aircraft entering the taxiway from any of the gates has a current position smaller than that of this third aircraft. Here it is important to remember that there is not an aircraft in third position on the taxiway since there
7.2 The taxiways in the model

are only two aircraft on the taxiway. However, there must be properties for every possible place in the line, even though there is not an aircraft in that place. Therefore, the current position property of that place in the line must be such that it will not prevent any aircraft from entering the taxiway. The position of this aircraft is set to 1 000 to establish this. This is done for aircraft moving in the direction of gate 1. If the aircraft are moving in the direction of gate 17, the position of the flight that was the last aircraft in the queue before another aircraft left the taxiway, must be set to zero. This is to ensure that no aircraft entering the taxiway have a smaller current position than this last aircraft, since this aircraft has shifted to the place of one less than that and the previous place at which the aircraft was is now empty. This previous place must thus not appear to be occupied by an aircraft and therefore prevent another aircraft from entering the taxiway. This is done automatically in the above described logic since the properties of the aircraft in fourth place in the line (which is not actually there since there were only three aircraft in the line) that are assigned to the aircraft in third place (which is also not there anymore) include a value for current position of 1 000 for aircraft travelling in the direction of gate 1, or zero for aircraft travelling in the direction of gate 17. These are the default properties for the places in the line that are not occupied by aircraft.

The logic described above can be coded as in Algorithm 3:

**Algorithm 3** Logic for an aircraft exiting a taxiway

\[\text{counter } 1 = \text{The place in line of aircraft B}\]

While \(\text{counter } 1 \leq \text{number of aircraft on taxiway}\)

\[\text{place in line (counter } 1) \leftarrow \text{place in line (counter } 1 + 1)\]

\[\text{position of aircraft (counter } 1) \leftarrow \text{position of aircraft (counter } 1 + 1)\]

\[\text{gate of aircraft (counter } 1) \leftarrow \text{gate of aircraft (counter } 1 + 1)\]

\[\text{counter } 1 \leftarrow \text{counter } 1 + 1\]

end

When \(\text{counter } 1\) equals the number of aircraft on the taxiway, \(\text{counter } 1 + 1\) will represent the place in the line of the aircraft at the place equal to one
more than the number of aircraft on the taxiway. It will therefore represent an unoccupied place in the line. The properties of an aircraft at this place (thus an aircraft that is not actually there) include the value for current node of 1 000 for aircraft taxiing in the direction of gate 1. This property is then assigned to the place in the line equal to the number of aircraft on the taxiway. Therefore, when an aircraft leaves the taxiway, this property for current node is assigned to the last place in the line, since there will then be one less aircraft and the last aircraft is not there anymore.

### 7.3 The speed at which the aircraft travel

The maximum allowable speed for aircraft according to van Ravesteyn (2011) and Rademan (2011) is as follows:

- on taxiways outside the ramp area: 30 knots (55.56 km/h)
- on taxiways inside the ramp area: 15 knots (27.78 km/h)
- for a 90 degrees turn: 10 knots (18.52 km/h)
- for entering a gate: 4 knots (7.41 km/h)

These are the aircraft speeds used in the simulation models. Since the runway processes were not modelled, the maximum allowable aircraft speed on the runway is not considered.

### 7.4 Loading and unloading of passengers

As soon as an aircraft arrives at the gate, the aircraft and the arriving passengers are separated and the passengers start moving towards the concourse while the aircraft stays at the gate. When boarding time for the departing passengers is reached, these passengers start moving through the concourse to the aircraft where it is parked at the gate. Once they arrive at the gate, they are combined with the aircraft. As soon as the time it takes to combine the aircraft and the departing passengers has elapsed, the aircraft is ready to taxi away from the gate.
7.5 Passenger paths in the concourses

The time it takes for boarding of the passengers on the different aircraft types is summarised below:

- for aircraft of types A, B and C: 10 minutes (S.A.S. Airbus, 2011a)
- for aircraft of types D: 14 minutes (Boeing, 2010)
- for aircraft of types E and F: 26 minutes (S.A.S. Airbus, 2011b)

The model uses these times for different aircraft types to combine the aircraft and passengers.

7.5 Passenger paths in the concourses

As stated in section 5.2, there will not be an automatic people mover at the new Lanseria International Airport like there is at Atlanta International Airport. There will, however, be pedestrian walkways as well as enough space to walk next to the pedestrian walkways. Passengers will thus have an option to choose either one. Young (1999) states in his article on pedestrian walking speeds in airport terminals that passengers using the pedestrian walkways actually have a lower average walking speed than those walking freely. This is due to the fact that many passengers stand on the walkways instead of walking. Also, these passengers then become an obstruction for passengers that do want to walk on the walkways, who are then forced to walk slower.

Fruin (1971) stipulates the walking speeds for passengers in an airport. For passengers using the pedestrian walkways the walking speed is 3.7 km/h and for passengers walking freely the walking speed is 4.8 km/h. These walking speeds were used in the simulation models.

7.6 Concluding remarks on chapter 7

The goal of developing the simulation models is to compare the different airport layouts. The logic in the simulation models, that were discussed in this chapter, is illustrated in the concept model in Figure 7.15. Processes and decisions
7.6 Concluding remarks on chapter 7

made in the models are explained in this concept model. The translation of the concept model to a computer model was discussed in detail in this chapter. The creation of the entities, the movement of aircraft on the taxiways and processes designed to avoid collisions of aircraft, the aircraft speeds used in the model, the loading and unloading of passengers onto the aircraft and the movement of the passengers from the terminal building to the boarding gates were explained. The models can now be altered to fit the different scenarios in terms of airport layout and to incorporate the different rules for assigning flights to gates. In order to ensure that the models were built correctly and that realistic comparisons can be made, extensive validation and verification processes were performed. This will be explained in chapter 8.
7.6 Concluding remarks on chapter 7
Chapter 8

Validation and verification of the simulation models

In order to ensure that the model is a realistic representation of the real world scenario, it must be verified and validated. Verification can be done by asking: “Was the model built correctly?” Thus, in verifying the model, the logic is inspected, the syntax errors are corrected and the runtime errors are also corrected. Validation is performed by asking: “Was the correct model built?” In other words, is the model a sufficient representation of the actual system? The
8.1 Validation and verification techniques used

Verification process is mostly executed together with the validation process since the model has to be built right in order to test whether it is realistic (Du Plessis, 2008).

Banks (1998) and Law & Kelton (2000) describe several validation and verification techniques. These techniques are used to validate the simulation models. The application of these techniques in the study is discussed in this chapter.

8.1 Validation and verification techniques used

The process of designing the model is described below:

- At first the creation of the different types of entities was modelled. It was ensured that the entities were created at the correct time, which is the arrival time, read from the schedule for the aircraft and the arriving passengers and the boarding time for departing passengers. Since only one entity represents all the arriving passengers and one entity the departing passengers, it was important to verify that the property namely number of passengers, which equals the number of arriving/departing passengers on each flight, was correct for each representative passenger. Also, the aircraft and the departing passengers had to move to the correct gate as calculated by the model. This was tested by assigning an aircraft and the departing passengers for its next flight to a specific, known gate and observing whether the aircraft did in fact go to that gate.

- Only one gate was designed having all the properties each gate should have, for example the gate size and the speed of aircraft coming into the gate. The processes involved at each gate were incorporated and tested to ensure that the loading and unloading of passengers were performed correctly.

- After the first gate was verified and validated, more gates were included. The total number of gates at that stage was five. The taxiways between the gates were modelled and it was ensured that the aircraft entered and exited the taxiways in a safe manner without causing two or more aircraft to collide.
8.1 Validation and verification techniques used

- The first concourse was then designed. The taxiway now had gates to which the aircraft could exit on both sides. The model was then validated specifically to prevent two aircraft at gates opposite each other, at the same position on the taxiway, to exit the gates at the same time, since this would cause them to crash once they got onto the taxiway.

- The pedestrian walkways and space for walking freely inside the concourse were then designed. It was ensured that passengers walked at the correct speed on the walkways and the open space and also that there were paths connecting the main terminal building with the gate to which they had to go.

- After the first concourse was modelled and validated, the other concourses were added. The taxiways between the concourses were modelled and the processes to avoid collisions were designed.

In order to uncover potential problems in the models after the whole layout was designed, a very busy flight schedule was used. When the model was run with this schedule, there were not enough gates to accommodate all the flights. Aircraft now had to wait to be assigned to gates. This was done to see if the model could handle this large number of aircraft and passengers. One problem that was found by using this method was that some of the gates were designed so that the aircraft never stopped to unload the passengers and wait for the departing passengers, because the paths connecting the taxiway with the gates were connected to the wrong nodes. Thus the aircraft left the gate as soon as it arrived. This mistake could then be corrected. Furthermore, the model was run using a very relaxed schedule. In using the first and second rules of assigning flights to gates, most of the flights were supposed to be assigned to the gates closest to the terminal building since very few gates were necessary for such a relaxed schedule and most gates would be available when a flight arrived. The model was run and this process was tested.

Animation was greatly used as a validation technique. By using animation, it could be observed whether aircraft collided, whether aircraft were directed to the correct gates and whether they spent the correct amount of time at the gates.
8.1 Validation and verification techniques used

8.1.1 Model reasonableness

Law & Kelton (2000) introduced the verification and validation factors which were incorporated as described below:

- **Continuity**: good continuity was shown in the model since the difference in input was reflected in the output. When a busier flight schedule was used, the time spent in the system by arriving passengers and aircraft was increased since they had a larger probability of having to wait for a gate to become available. Also, if an aircraft had to wait for a gate, chances were it would be delayed since the unloading process started late. The delay in the flight arrival would cause the departing passengers to be delayed. This was also reflected in the output since the time these passengers spent in the system, using a busier schedule, increased. When a relaxed schedule was used, the average walking distance per passenger was lower than when a busier schedule was used. This is due to the fact that fewer aircraft had to park at gates far away from the main terminal, since the gates near the terminal would usually be available.

- **Degeneracy**: when the number of gates at the airport was reduced but the same flight schedule was used, the number of aircraft waiting for a gate to become available increased, as did the time the aircraft and the passengers spent in the system. This had the same effect as when the number of gates was kept the same, but the schedule was busier.

- **Absurd conditions**: when the model was run with only one gate, or with an outrageously busy schedule, not all flights could be processed and the airport got far behind schedule with aircraft and passengers piling up in the system.

- **Consistency**: simulation runs using slightly different flight schedules produced similar results.
8.1 Validation and verification techniques used

8.1.2 Face validation

Face validation was used to ensure correctness of the model through inspection and experimentation and to identify problems. Table 8.1 summarises the face validation process done in the verification and validation process.

<table>
<thead>
<tr>
<th>Function</th>
<th>Criteria</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating arriving passengers</td>
<td>Does the representative arriving passenger have the correct property of number of passengers?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Are the arriving passengers created at the correct time according to the schedule?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Does the number of input arriving passengers equal the number of output arriving passengers?</td>
<td>YES</td>
</tr>
<tr>
<td>Creating departing passengers</td>
<td>Does the representative departing passenger have the correct property of number of passengers?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Are the departing passengers created at the correct boarding time according to the schedule?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Are the groups of departing passengers created in the correct order according to the boarding times?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Does the number of input departing passengers equal the number of output departing passengers?</td>
<td>YES</td>
</tr>
<tr>
<td>Creating aircraft</td>
<td>Are the aircraft created at the correct time according to the schedule?</td>
<td>YES</td>
</tr>
</tbody>
</table>

Continued on next page
### 8.1 Validation and verification techniques used

#### Table 8.1 – continued from previous page

<table>
<thead>
<tr>
<th>Function</th>
<th>Criteria</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time according to the schedule?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Are the aircraft combined with the correct group of arriving passengers?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Does every aircraft have the correct flight number/priority?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Is every aircraft of the correct type according to its wingspan?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Do the different aircraft arrive in the correct order?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Are the aircraft all assigned to suitable gates according to their wingspan?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Do the aircraft depart at the correct departure time?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Does the number of input aircraft equal the number of output aircraft?</td>
<td>YES</td>
</tr>
<tr>
<td>Taxiing of aircraft</td>
<td>Does each aircraft taxi to its assigned gate?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Do all aircraft wait their turn before entering a taxiway?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Is the collision of aircraft avoided at all times?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Do all aircraft taxi at the correct speed depending on the type of taxiway?</td>
<td>YES</td>
</tr>
<tr>
<td>Passenger walking</td>
<td>Do all passengers walk at the correct</td>
<td>YES</td>
</tr>
</tbody>
</table>

Continued on next page
8.2 Concluding remarks on chapter 8

In order to ensure that the models were built correctly and that they can be compared realistically, different verification and validation techniques were applied. This was explained in this chapter. The process of designing the model, by starting with a small part (one gate) and validating it before expanding the airport, was discussed. The incorporation of validation and verification factors such as continuity, absurd condition test, degeneracy and consistency was explained as well as the face validation process. The models were found to be valid and can thus be used to compare the different scenarios as will be done in chapter 12. Since all the results obtained in the study will be provided in chapter 12, the metaheuristic optimisation part of the problem solving phase will be discussed.

---

### Table 8.1 – continued from previous page

<table>
<thead>
<tr>
<th>Function</th>
<th>Criteria</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gates</td>
<td>Is only one aircraft assigned to a gate at any time?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Does each aircraft spend the correct amount of time at the gate depending on the departure time of its next flight?</td>
<td>YES</td>
</tr>
<tr>
<td>Loading passengers</td>
<td>Is the correct group of departing passengers loaded onto the aircraft?</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Are the aircraft and departing passengers combined at the correct boarding time?</td>
<td>YES</td>
</tr>
<tr>
<td>Unloading passengers</td>
<td>Are the aircraft and arriving passengers separated as soon as the aircraft arrives at the gate?</td>
<td>YES</td>
</tr>
</tbody>
</table>
8.2 Concluding remarks on chapter 8

before the results of the simulation models will be given. The metaheuristic optimisation part is started in chapter 9 in which an overview of metaheuristic techniques will be provided and the cross-entropy method will be explained. This will be done in preparation for the discussion of the metaheuristic optimisation process developed for the purpose of assigning flights to gates with the objective of minimising the passenger walking distances at the airport.
Chapter 9

Objective optimisation by using the cross-entropy method

The concept of optimisation is used to analyse many complex allocation or decision processes. By using an optimisation technique, values for a set of interrelated variables in a complex decision problem are selected by focusing on an objective designed to improve the quality of the process. The quality is measured by quantifying the performance of the process (Luenberger et al., 2008). The method of metaheuristics is a subfield of stochastic optimisation. Metaheuristics
9.1 Different metaheuristic techniques

are algorithms used to solve problems and optimise objectives when the search space is too large to solve the problem analytically (Visser, 2010). In this chapter, different metaheuristic techniques are discussed with specific focus on the cross-entropy method.

9.1 Different metaheuristic techniques

Gendreau & Potvin (2010) discuss the use of metaheuristics and the different metaheuristic techniques. These techniques include, amongst many others, evolutionary algorithms, hill climbing, steepest ascend hill climbing, tabu search and simulated annealing.

Evolutionary algorithms are well-known techniques for solving objective optimisation problems. In evolutionary algorithms natural selection amongst individuals in a population is caused by pressures in the environment, as can be described by survival of the fittest. The result is that the fitness of the population rises. A quality function exists and must be maximised, thus a random set of elements from the function is created to which this quality function is applied as a fitness measure. Elements that are better than the other are combined (such as two parents producing one child) or mutated (such as one parent producing a child on his own). These children then compete with older elements on the basis of fitness to become part of the next generation. This process is repeated until either the limit of the computation is reached or an element with a sufficient quality is found (Eiben & Smith, 2003).

Goldberg introduced the use of a niche, such as fitness sharing (Coello Coello, 2006). Fitness sharing is to lower the fitness of each element in the population by an amount that is close to the number of elements that are similar in the population. The reason for using a niche is to minimise the effects of genetic drift that may result during the selection process. Furthermore, it aids in maintaining the diversity of the population and in preventing the genetic algorithm from getting trapped in local optima (Saren & Krähenbühl, 1998).

When using the hill climbing method, a random solution is generated and then tweaked. If the new solution is better than the original one, the original solution is replaced by the new solution. This process is continued until the ideal
9.1 Different metaheuristic techniques

solution has been discovered or until a specified number of iterations has been performed. To tweak a vector solution, a small amount of random noise can be added to each number in the solution (Visser, 2010).

The steepest ascend hill climbing method is similar to the hill climbing method, however, instead of applying only one tweak to the candidate solution, many tweaks are applied at the same time. The best solution of all the tweaked solutions including the original, is then selected for the next iteration (Luke, 2009).

In the tabu search method, a history is kept in the tabu list of candidate solutions that were recently considered. None of these solutions are considered again until they are sufficiently far in the past. A solution (S) is created and placed on the tabu list. If the tabu list is full, the oldest solution in the list is removed. The new solution is then tweaked a few times and the best of the tweaked solutions (R) is placed on the tabu list. If this best solution (R) is better than the original solution (S) it replaces the original solution (S) to be tweaked in the next iteration. A record is kept of the best solution so far (Visser, 2010).

When using simulated annealing the variable \( t \) (temperature) is assigned a high value. A random feasible solution (S) is then created. The created solution is tweaked as in the hill climbing method. If the quality of the first solution (S) is worse than that of the new, tweaked solution (R), it is replaced by the new solution. The new solution is then used in the next iteration. The way in which the quality of the solution is tested is shown in equation 9.1:

\[
\text{Random number between 0 and 1} < e^{\frac{\text{Quality}(R) - \text{Quality}(S)}{t}}
\]  

(9.1)

Here, if R is better than S, the fraction will be positive and the probability will be larger than one and S will always be replaced by R. However, if R is worse than S (thus the new solution is worse than the original one) then the fraction is negative. If R is far worse than S, the fraction is larger and the probability will be closer to zero. This means that the chance of the random number between zero and one being smaller than the calculated probability is slim and the probability for selecting the original solution, S (which is better), is big. However, if R is not much worse than S, the chance for selecting R, even though it is worse than S,
9.2 The cross-entropy method

is bigger. The larger the value of \( t \), the larger the fraction, and if the fraction is negative, the probability will be closer to zero. After every iteration the value of \( t \) is decreased (Visser, 2010).

The metaheuristic technique used in this study is the cross-entropy (CE) method. The reason for using the CE method is because it has been proven to converge quickly towards the optimum solution. The CE method is explained in section 9.2.

9.2 The cross-entropy method

An increasingly popular approach to solve complicated optimisation problems is through stochastic algorithms and specifically the cross-entropy method (CE) (Rubinstein & Kroese, 2004). At first, the CE method was used to estimate probabilities of rare events in networks that were complex and stochastic and which involved variance minimisation (Rubinstein, 1997). However, it was soon discovered that by applying simple modifications, the CE method could also be used to solve difficult combinatorial optimisation problems (Rubinstein, 1999).

The following applications, amongst others, have been found for the CE method:

- buffer allocation (Alon et al., 2005)
- static simulation models (Rubinstein & Kroese, 2004)
- queuing models of telecommunication systems (De Boer, 2000; De Boer et al., 2004)
- scheduling (Margolin, 2002)
- vehicle routing (Chepuri & Homen de Mello, 2005)

It must be noted that, as far as could be determined, the CE method has not yet been applied to improve the flight-to-gate assignment process at an airport. Furthermore, this method has never been applied in real time to simulation problems. However, in this study the CE method will be applied in real-time, it will thus be repeated throughout the simulation run to account for delays.
9.2 The cross-entropy method

9.2.1 Steps and procedures in the CE method

In the CE method, an iterative procedure is performed. The following steps are performed in each iteration (Rubinstein & Kroese, 2004):

- A random sample of solutions is generated based on a specific mechanism.

- The parameters of the mechanism are adjusted and updated according to the data in order to produce a better data sample in the following iteration.

The procedure in this method is described as follows (Bekker, 2011; De Boer et al., 2005):

- A sample of \( n \) random feasible solutions or population members is created. Here, \( n \) is the population size.

- The performance of each of these solutions is calculated.

- The solutions are sorted from worst to best based on the performance of each.

- A specified percentage of the solutions is selected from the bottom up, thus a specified percentage of the best solutions. These best solutions are then used to calculate the probability for each parameter in the solution having a specific value. The probabilities are thus adjusted in each iteration based on the solutions in the previous iteration with the best performances.

- These new calculated probabilities are then used in creating the new sample of \( n \) solutions in the next iteration.

- This process is repeated until the best or a good enough solution has been found.

The above explanation is a simple description of the CE method. The complex mathematical formulation of this method is briefly discussed in the next section.
9.2.2 Mathematical formulation of the CE method

The discussion of the mathematical formulation of the CE method is based on that by Rubinstein & Kroese (2004). “Importance Sampling” and the “Kullback-Leibler distance” form the foundation of the CE method for optimisation and are discussed in this section.

A random sample vector, $X = (X_1, \ldots, X_n)$ from some space $X$ (i.e. a collection of all possible solutions), has a real function $f$. In the flight-to-gate assignment optimisation, discussed in chapter 11, $X$ is a population consisting of a random sample of $n$ solutions. In each solution, $X_i$, all the flights in consideration are assigned to suitable gates. The real function, $f$, of each solution is the performance of that solution (i.e. the total passenger walking distance) that needs to be minimised.

The probability that the performance, $f(X)$, is smaller than or equal to a real number $\gamma$ under a family of probability density functions $h(\cdot; u)$ on $X$ needs to be determined. In this probability:

$$l = \mathbb{P}_u(f(X) \leq \gamma) = \mathbb{E}_u I_{\{f(X) \leq \gamma\}}, \quad (9.2)$$

if $l$ is very small, the probability that $f(X)$ is smaller than or equal to $\gamma$, i.e. $f(X \leq \gamma)$, is a “rare event”. Importance sampling can be used to efficiently estimate this probability. A random sample $X_1, \ldots, X_N$ is taken from a different density $g$ on $X$ for this purpose. The following likelihood ratio estimator is used to estimate the value of $l$ in this case (Rubinstein & Kroese, 2004):

$$\hat{l} = \frac{1}{N} \sum_{i=1}^{N} I_{\{f(X_i) \leq \gamma\}} \frac{h(X_i; u)}{g(X_i)}. \quad (9.3)$$

This can be written as:

$$g^*(x) = \frac{I_{\{f(x) \leq \gamma\}} h(x; u)}{l} \quad (9.4)$$

which means

$$l = \frac{I_{\{f(X) \leq \gamma\}} h(X; u)}{g^*(X)}. \quad (9.5)$$
9.2 The cross-entropy method

Here, \( g^* \) is an estimation of the optimum and depends on the unknown value of \( l \). However, the value of \( g^* \) can be approximated within the family of densities \( \{ h(\cdot; \mathbf{v}) \} \). The reference parameter in this case is \( \mathbf{v} \). The approximation of the value of \( g^* \) is done by minimising the distance between \( g^* \) and \( h(\cdot; \mathbf{v}) \). The Kullback-Leibler distance or cross-entropy between \( g \) and \( h \) is the measure of this distance, and it is defined as:

\[
\mathcal{D}(g, h) = \mathbb{E}_g \ln \frac{g(X)}{h(X)} = \int g(x) \ln g(x) dx - \int g(x) \ln h(x) dx.
\]

For the distance between \( g^* \) in Equation (9.4) and \( h(\cdot; \mathbf{v}) \) to be minimised, the value of \( \mathbf{v} \) must be selected in a way that will minimise \(- \int g^*(x) \ln h(x; \mathbf{v}) dx\). This minimisation can be achieved by solving the following maximisation problem:

\[
\max_{\mathbf{v}} \int g^*(x) \ln h(x; \mathbf{v}) dx.
\]

The replacing of \( g^* \) of Equation (9.4) in Equation (9.8), results in the following maximisation formulation:

\[
\max_{\mathbf{v}} \int I_{\{f(X) \leq \gamma\}} h(X; \mathbf{u}) \ln h(X; \mathbf{v}) dx.
\]

This corresponds to:

\[
\max_{\mathbf{v}} \mathbb{E}_\mathbf{u} I_{\{f(X) \leq \gamma\}} \ln h(X; \mathbf{v}).
\]

When considering this discussion of Importance Sampling and the Kullback-Leibler distance, optimisation by using the CE method can be formulated. In this explanation of the CE method, the minimum value of \( f(x) \) (the performance function) over all states \( x \) in some set \( \mathcal{X} \) needs to be determined. Let the minimum value of \( f(x) \) be \( \gamma^* \), then

\[
\gamma^* = \min_{x \in \mathcal{X}} f(x).
\]
9.2 The cross-entropy method

\{h(\cdot; v), v \in \mathcal{V}\} is a defined family of probability density functions on the set \(X\). The following estimation problem is the stochastic problem associated with Equation (9.11):

\[
l(\gamma) = \mathbb{P}_u(f(X) \leq \gamma) = \mathbb{E}_u I_{\{f(X) \leq \gamma\}}.
\]  

(9.12)

The random vector, \(X\), has a probability density function, \(h(\cdot; u)\) for some \(u \in \mathcal{V}\). The probability of \(\{f(X) \geq \gamma\}\) when estimating \(l\) is a rare event. By using the Kullback-Leibler cross-entropy, the estimation of \(l\) can be done through making adaptive changes to the probability density function. A number of probability density functions \(h(\cdot; u), h(\cdot; v_1), h(\cdot; v_2), \ldots\) is created in sequence. Thus, a series of tuples \(\{(\hat{\gamma}_t, \hat{v}_t)\}\) is generated that converges to the optimal tuple \((\gamma^*, v^*)\). By setting \(v_0 = u\), the procedure in each iteration is as follows (Rubinstein & Kroese, 2004):

1. **Adaptive updating of \(\gamma_t\).** For \(v_{t-1}\) which is fixed, let \(\gamma_t\) be the \((1 - \varrho)\)-quantile of \(f(X)\) under \(v_{t-1}\). That is, \(\mathbb{P}_{v_{t-1}}(f(X) \leq \gamma_t) \geq \varrho\). \(\varrho\) is typically chosen as \(\varrho = 10^{-2}\). \(\gamma_t\) must now be estimated by drawing a random sample \(X_1, \ldots, X_N\) from \(h(\cdot; v_{t-1})\) and determining the \((1 - \varrho)\)-quantile of the sample performances

\[
\hat{\gamma}_t = f_{\lfloor(1-\varrho)N\rfloor}.
\]

(9.13)

2. **Adaptive updating of \(v_t\).** The following program must be solved in order to derive \(v_t\) for \(\gamma_t\) and \(v_{t-1}\):

\[
\min_v D(v) = \min_v \mathbb{E}_{v_{t-1}} I_{\{f(X) \leq \gamma_t\}} \ln h(X; v).
\]

(9.14)

The stochastic program

\[
\min_v \hat{D}(v) = \min_v \frac{1}{N} \sum_{i=1}^{N} I_{\{f(X_i) \leq \hat{\gamma}_t\}} \ln h(X_i; v)
\]

(9.15)

can be used to estimate \(\min_v D(v)\) in Equation (9.14).

The following smoothing function can be used to update the parameter vector \(\hat{v}\):

\[
\hat{v}_t = \alpha \hat{v}_t + (1 - \alpha) \hat{v}_{t-1}
\]

(9.16)
9.3 Concluding remarks on chapter 9

where $\hat{v}_t$ is obtained from Equation (9.15). In the flight-to-gate assignment optimisation that will be discussed in chapter 11, $v_t$ is the set of probabilities of each flight being assigned to each gate. These probabilities are adjusted in each iteration of the CE method based on the solutions that resulted in the smallest passenger walking distances in the previous iteration. Algorithm 4 shows the CE method for optimisation by Rubinstein & Kroese (2004), based on the discussion above.

Algorithm 4 Main CE Algorithm

1: Choose some $\hat{v}_0$. Set $t=1$.
2: Generate a sample $X_1, \ldots, X_N$ from the density $h(\cdot; \hat{v}_{t-1})$ and compute the sample $(1-\varrho)$-quantile $\hat{\gamma}_t$ of the performances according to Equation (9.13).
3: Use the same sample $X_1, \ldots, X_N$ and solve the stochastic program in Equation (9.15). This solution is $\tilde{v}_t$.
4: Smooth the vector $\tilde{v}_t$ using Equation (9.16).
5: If for some $t \geq d$, say $d = 5$, $\hat{\gamma}_t = \hat{\gamma}_{t-1} = \ldots = \hat{\gamma}_{t-d}$, then stop, otherwise set $t \leftarrow t + 1$ and return to Step 2.

9.3 Concluding remarks on chapter 9

In this chapter, metaheuristic techniques such as genetic algorithms, simulated annealing, and the cross-entropy method were briefly explained. The technique used in this study is the cross-entropy method, and the applications thereof were discussed in this chapter. Furthermore, the steps and procedures followed in the cross-entropy method and the mathematical formulation thereof were explained. A good understanding of this technique was developed and can be used to design a metaheuristic optimisation process for assigning flights to gates in order to reduce the passenger walking distances. Before the optimisation process designed for this purpose will be discussed, two case studies in this regard will be presented in chapter 10.
Chapter 10
Gate assignment operation case studies

In an airport gate assignment operation, aircraft must be assigned to gates in a way that meets the operational requirements and ensures minimum inconvenience to all passengers. Passenger inconvenience can be reduced by minimising the walking distance from check-in to the departure gates, from the gates to the luggage claim area as well as between gates for transfer passengers. Minimising the latter distance is crucial for preventing delays between flights with short con-
A case study by Yan et al.

Aircraft must be assigned to gates that will allow passengers to board/deplane in the most convenient way. These assignments influence the efficiency of the airport as well as its level of service (LOS). Before the study done by Yan et al. (2002), several models have been developed to plan static gate assignments. However, none of these considered stochastic flight delays that occur in real-time operations. Yan et al. (2002) investigated the interrelationships that occur between static and real-time gate assignments of which the latter is affected by flight delays. The study also incorporates buffer times in the gate assignments that absorb delays that occur in real-time operations.

The simulation done in the study is divided into a planning part and a real-time part. A mathematical programming model and a solution algorithm are proposed in the planning part to optimise the assignment process. Furthermore, two heuristics are proposed to do a comparison and find good solutions. In the real-time part of the simulation, flexible buffer times, rules of the real-time reassignment and flight delay patterns are suggested.

After the required information has been gained and buffer times have been included in the simulation process, the optimisation model and the two proposed heuristics are used to solve the static gate assignment problem. Next, the departure and arrival times for each flight are generated after incorporating the flight delays distribution. In the real-time simulation, an aircraft is reassigned to another gate if it cannot occupy its original gate. After the simulation is conducted for one day, the effect of flight delays on static gate assignments is evaluated.
The information necessary to do the simulation includes:

- the layout of the airport terminal and the number of gates
- the flight schedule for a day in terms of arrival and departure times of each flight
- the type of aircraft for each flight
- the number of departing, arriving and transferring passengers on each flight
- the buffer time which has been set at 30 minutes for international flights of which 15 minutes are incorporated before a flight’s time window and 15 minutes after
- the walking distance for each arriving, departing and transfer passenger to and from the gate
- the performance criteria is the total walking distance of all passengers

An assumption is made that, since the maximum flow cannot exceed the capacity of the airport, each flight must be assigned to a gate. In other words, temporary aircraft holding until a gate becomes available does not apply.

The formulation of the static gate assignment problem is as follows:

\[
\begin{align*}
\min Z &= \sum_{i=1}^{M} \sum_{j=1}^{N} c_{ij} x_{ij} \\
\text{s.t} \sum_{j=1}^{N} x_{ij} &= 1 \text{ for all } i \\
\sum_{i \in L_s} x_{ij} &\leq 1 \text{ for all } s, \text{ for all } j \\
x_{ij} &= 0 \text{ or } 1 \text{ for all } i, \text{ for all } j
\end{align*}
\] (10.1)

Here, \( M \) denotes the total number of flights, with \( i \) the \( i^{th} \) flight and \( N \) the total number of gates, with \( j \) the \( j^{th} \) gate. \( Z \) denotes the total walking distance of the passengers. The set of flights is represented by \( L_s \). \( c_{ij} \) is the distance that passengers on flight \( i \), that is assigned to gate \( j \), must walk. If flight \( i \) may not
10.1 A case study by Yan et al.

be assigned to gate \( j \), \( x_{ij} \) is eliminated. \( x_{ij} \) is equal to one if flight \( i \) is assigned to gate \( j \) and zero otherwise.

The constraints in (10.2) ensure that each flight is assigned to no less and no more than one gate. The constraints in (10.3) denote that no more than one flight can be assigned to each gate at the same time. This problem can be solved with the simplex method together with the branch and bound technique. The following two heuristics were developed in order to do a comparison with the real-time optimal solutions:

- The first heuristic tries to minimise the total passenger walking distance. The flights are sorted in decreasing order based on the number of passengers on the flight. It then assigns the flights to the closest gates that are not yet occupied.

- The second heuristic works in the exact same way, except the flights are now sorted in increasing order based on arrival times.

In assigning flexible buffer times, the following rules were set:

- Buffer times may not exceed 30 minutes.

- Buffer times are changed with intervals of five or ten minutes.

- Shorter buffer times are used for high flight density period flights in order to minimise a reduction in the capacity of the airport.

- Longer buffer times are assigned to low flight density period flights in order to minimise disturbances in flight assignments.

- An assumption is made that the buffer times absorb flight delays.

Two types of solutions are used in the real-time assignment problem. The first, that does not allow the influence of other planned aircraft assignments (IOPAA), shows that any incoming flight may be reassigned to any gate if an overlapping planned incoming flight has not been assigned to it. The second shows that any incoming flight may be reassigned to any gate even if an overlapping incoming flight has been assigned to it.
10.1 A case study by Yan et al.

Three options have been developed that can be used to reassign incoming flights. These options are:

- using the gate with the minimum distance from the original gate
- using the gate with the minimum distance that is neighbouring the original gate
- using a random gate

Combining the two solution types and the three options, six ways of real-time gate assignments were formulated.

The static gate assignments were solved after the necessary test data were obtained. This showed that the objective value of the optimisation model was better than that of the two heuristics, of which the first seemed to be better than the second.

In an attempt to measure the effect of stochastic flight delays on static gate assignments, the following two indices were used:

\[
\text{Degree of reassignment (DR)} = \frac{\text{number of reassigned flights}}{\text{number of total flights}} \tag{10.5}
\]

\[
\text{Objective increase (OI)} = \frac{\text{real-time objective} - \text{static objective}}{\text{static objective}} \tag{10.6}
\]

The evaluation showed that as the variance of the flight delays increased, DR and OI increased. However, the increments in both these indices decreased as the variance increased. Furthermore, it was observed that the optimisation model was more affected by delays than the two heuristics. It is clear, when evaluating the DR and OI, that static gate assignments are indeed negatively affected by flight delays.

It was further observed that the DR, when the influence of other planned aircraft assignments was not allowed, was less than when this influence was allowed. The OI, however, was more when the influence was not allowed. A trade off exists between these two indices in this case. When analysing the three gate assignment options, it could be seen that in terms of DR, the random gate assignment was
the best, with the assignment based on minimum overall distance (option 1) is the worst. In terms of OI, the assignment based on the closest neighbour (option two) wins, with the random assignment (option 3) coming last.

10.2 A case study by Drexl and Nikulin

According to Drexl & Nikulin (2008), the problem of flight to gate assignments on an airport refers to assigning aircraft to their stand positions while satisfying a set of constraints. These stand positions of the aircraft are called the gates and passengers enter and exit the terminal building through them.

The following two restrictions are present in a gate assignment problem (GAP):

- Only one aircraft may occupy any gate at any time.
- Each flight is assigned to one and only one gate.

Further limitations occur when considering the comfort of passengers and the convenience for the airport services. An airport has the following goals when assigning flights to gates:

- minimising the number of flights not assigned to gates
- minimising the total passenger walking distance (or the passenger connection times)
- maximising the preference of assigning specific flights to specific gates

Referring to the first goal, if no gates are available for a flight to be assigned to, the aircraft is assigned to the apron. This results in a decrease in passenger comfort since they must now be transferred to the terminal in a bus. This increases passenger waiting times.

The second goal concerns the minimising of walking distances. These walking distances can be calculated as the distance from the gate to the baggage claim area for arriving passengers, the distance between the check-in area and the gates for departing passengers and the distance between gates for transferring passengers.
10.2 A case study by Drexl and Nikulin

Another distance, namely the distance from the terminal to the apron or from the apron to the terminal, is added when flights are assigned to the apron.

In the third goal, preference values are assigned to each combination of flights and gates and each gate receives a priority value. The flight gate preference score must be maximised in order to maximise airport convenience.

The problem is complex, since it has many constraints and multiple objectives. It is thus unlikely that a single solution that optimises all objectives will be found. It is required to find a solution that will produce an acceptable value for all the objectives, while satisfying all the constraints. Input data for this problem includes:

- the time table of arrivals and departures for all flights
- the origins of all flights
- the destinations of all flights
- the aircraft type of each flight
- the number of passengers on each flight
- a flight is either international or domestic
- the gate preference for each flight

This GAP can be formulated as shown below:

\[
N = \text{set of all flights at the airport (arriving as well as departing)}
\]
\[
M = \text{set of all gates at the airport}
\]
\[
n = \text{the number of flights, } n = |N|
\]
\[
m = \text{the number of gates, } m = |M|
\]
\[
a_i = \text{the time of arrival of flight } i
\]
\[
d_i = \text{the time of departure of flight } i
\]
\[
w_{k,l} = \text{the distance to be walked between gates } k \text{ and } l
\]
\[
f_{i,j} = \text{the number of passengers coming from flight } i \text{ transferring to flight } j
\]
10.2 A case study by Drexl and Nikulin

\[ u_{i,k} = \text{the value of preference for assigning flight } i \text{ to gate } k \]

\[ v_i = \text{the priority of flight } i \]

Two extra gates are used in the formulation, gate 0, denoting the entrance (or exit) of the airport and gate \( m + 1 \), denoting the apron.

\[ \pi_{i,k} = 1 \text{ if flight } i \text{ is assigned to gate } k \text{ and 0 otherwise, } 0 < k \leq m + 1 \]

Objectives:

\[
\begin{align*}
\min z_1 &= \sum_{i=1}^{n} \pi_{i,m+1} \\
\min z_2 &= \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m+1} f_{i,j} w_{k,l} \pi_{i,k} \pi_{j,l} + \sum_{i=1}^{n} \sum_{k=1}^{m+1} f_{0,i} w_{0,k} \pi_{i,k} \\
&\quad + \sum_{i=1}^{n} \sum_{k=1}^{m+1} f_{i,0} w_{k,0} \pi_{i,k} \\
\max z_3 &= \sum_{i=1}^{n} \sum_{k=1}^{m+1} v_i u_{i,k} \pi_{i,k}
\end{align*}
\]

Objective 10.7 minimises the number of flights assigned to the apron. The second objective, 10.8, minimises the total passenger walking distance for transferring, departing and arriving passengers and objective 10.9 maximises the gate assignment preference value.

Constraints:

\[
\begin{align*}
\sum_{k=1}^{m+1} \pi_{i,k} &= 1, \quad 1 \leq i \leq n \\
\pi_{i,k} \pi_{j,k} (d_j - a_i) (d_i - a_j) &\leq 0, \quad 1 \leq i, j \leq n, \quad k \neq m + 1 \\
\pi_{i,k} &\in \{0, 1\}, \quad 1 \leq i \leq n, \quad 1 \leq k \leq m + 1
\end{align*}
\]

Constraint 10.10 must be satisfied to prevent a flight from being assigned to more than one gate. The second constraint, 10.11 ensures that only one flight is assigned to each gate at any time – no overlapping may occur. Constraint 10.12 defines the variables to be Boolean. This model does not explicitly include buffer times between assignments, however, it can easily be incorporated by changing the arrival and departure times.
One way of solving the problem in this case study is simulated annealing (SA). In this approach, an approximation of the optimum solution is located in the search space. A few iterations are completed in which neighbours of the approximate solution, \( \pi \), are considered. Either of these neighbours, \( \pi' \), can be accepted or \( \pi \) can be used. If \( \pi' \) has a better objective function value, it is chosen. When a “good enough” solution has been obtained or when a time limit has been reached, the process is stopped.

In each iteration, \( \pi' \) is accepted with the following probability:

\[
P(\pi, \pi', T_q) = \min\{1, e^{-\frac{z(\pi') - z(\pi)}{T_q}}\}.
\]

During the execution of the algorithm, the probability decreases. This is to guarantee that the global optimum will be located with high probability and to avoid getting trapped in a local optimum space.

The probability depends on \( T \) the annealing temperature, \( (T_q)_{q=1}^{\infty} \) with \( \lim_{q \to \infty} T_q = 0 \). \( T_0 \) must be defined. Shortly before the end of the process, \( T_q \) decreases to zero.

In a multiobjective problem, values for a set of variables that will allow optimisation of the objective functions are needed. \( \pi \) can be defined as a Pareto optimal solution if it is non-dominated by any other solution \( \pi' \). Then the set of all non-dominated solutions are the Pareto set. Pareto Simulated Annealing (PSA) can thus be used to solve this GAP.

The most important objective is to minimise the number of flights assigned to the apron. Thus, the first objective is optimised as the first step and then PSA is applied to the other two objectives. In obtaining the best possible solution for the first objective, a greedy strategy is used. Here, flights are sorted based on increasing departure times after which they are assigned to the gates one by one as long as there are gates available. In the case where no gates are available, aircraft are assigned to the apron (gate \( m + 1 \)).

The result obtained from the greedy strategy is the number of flights that could be assigned to gates as well as an initial solution \( \pi \) that can be used as a starting point for the PSA in solving the other two objectives. The first objective can now be ignored, since it is optimised and fixed.
Now, any solution that can be chosen starting from \( \pi \) in one move forms part of the neighbourhood. One way of selecting new solutions is through the interval exchange move. In this method, any flight assigned to gate \( k_0 \) can only be swapped with a flight assigned to \( l_0 \) if this move will not result in conflict between other flights due to the overlapping of scheduled times at these gates. This approach starts with two flights that are assigned to different gates but that overlap. If these two flights cannot be exchanged in the interval available, the interval of the adjacent flight is added in order to extend the initial interval. This will either allow an exchange between the two flights or the exchange will be impossible. Flights assigned to the apron can also be exchanged with flights assigned to gates.

### 10.3 Concluding remarks on chapter 10

From the two case studies discussed in this chapter it is clear that the use of metaheuristics, whether it is for single or multi-objective optimisation, improves or optimises the objectives of an airport. In the first case study, single objective optimisation is used to improve the passenger walking distances at the airport, and in the second case study, multi-objective optimisation is used. In this case study the objectives are the passenger walking distances that must be minimised, the number of flights assigned to the apron (and not to gates connected to the terminal building) that must also be minimised and the preference for assigning specific flights to specific gates that must be maximised. In chapter 11 the metaheuristic optimisation process, designed to assign flights to gates in a way that will reduce the passenger walking distances (Rule 3), will be discussed. Furthermore, a comparison between the two case studies and the metaheuristic developed in this study will be made.
Chapter 11

Assigning flights to gates using the cross-entropy method

A metaheuristic is introduced for applying Rule 3 of assigning arriving flights to gates, in order to optimise the total passenger walking distance. The method used for this purpose is the cross-entropy (CE) method. When the CE method is not used (as in Rule 1 and Rule 2), a flight is assigned to the gate that is the best for that specific flight, without considering the flights that will arrive after it. However, in Rule 3, a number of flights that will arrive after the flight in
question, is considered and each flight is assigned to a gate in a way that will be better for all the flights in consideration, in terms of passenger walking distances.

In the models using Rule 3 for assigning flights to gates, the metaheuristic optimisation, using the CE method, is executed in real-time. It is therefore repeated at certain times during the simulation run for various reasons that will be explained. When the metaheuristic optimisation is executed, the airport simulation is put on hold while the new flight-to-gate assignment is being calculated. As soon as the optimal solution has been found, the airport simulation is continued, using this optimal flight-to-gate assignment schedule. In this chapter, the metaheuristic optimisation is explained and will be referred to as “the optimisation process”. The airport simulation, that is put on hold while the new flight-to-gate assignment is being performed, will be referred to as “the main simulation” or as “the simulation”.

Each time the optimisation process is repeated the passenger walking distance for the flights considered in that execution is minimised. Several iterations are performed in each execution of this process in order for the solution to converge to the optimum. As described in section 9.2, a population consisting of a random sample of feasible solutions is created in each iteration. In each solution, all the flights in consideration are assigned to suitable gates. The CE method, in which several similar iterations are performed, is then applied to the created population. These processes are described by Figure 11.1. In each execution of the optimisation process in this figure, there are n iterations and in each iteration, there are m solutions in the population.

However, before the population of solutions in the first iteration of each execution of the optimisation process can be created, the state of the airport at the time of execution (for instance whether each gate is currently occupied or available) must be assessed. The optimisation process in Rule 3 of assigning flights to gates can thus be divided into four sections:

- **Overview of the optimisation process.** In this section, factors considered in the optimisation process and the background and feasibility of the application of the CE method are discussed.
Frequency of performing the optimisation process. As stated before, the CE method is applied in real-time. This will be explained in section 11.2. The reasons for and the frequency of performing the optimisation process is discussed in this section.

Actions performed only once. The actions in this section of the optimisation process are performed in preparation for creating the population of feasible solutions. The current state of the airport, each time the optimisation process is performed, is assessed and recorded. This assessment includes determining which is the next flight to arrive at and which is the next flight to depart from the airport, as well as evaluating the state of each gate (occupied/available).

Creating the population. Each time the optimisation process is performed, a number of iterations is executed. In each iteration a population of feasible solutions is created. Each solution stipulates to which gates the flights are assigned. The solutions can be created after the current state of the airport has been assessed and recorded. In this section of the

<table>
<thead>
<tr>
<th>First execution of optimisation process</th>
<th>Last execution of optimisation process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration 1</td>
<td>Iteration 1</td>
</tr>
<tr>
<td>Solution 1</td>
<td>Solution 1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Solution (m)</td>
<td>Solution (m)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Iteration (n)</td>
<td>Iteration (n)</td>
</tr>
<tr>
<td>Solution 1</td>
<td>Solution 1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Solution (m)</td>
<td>Solution (m)</td>
</tr>
</tbody>
</table>

Simulation time

Figure 11.1: The execution of the optimisation process
11.1 Overview of the optimisation process

optimisation process, the detail of the way in which each population member/solution is created is discussed. Since several flights are assigned to each gate over time, the processes of occupying and releasing the gates are addressed. These processes are complicated since the system changes over time and the state of each gate changes from available to occupied and vice versa. The calculation of the performance of each solution, i.e. the total passenger walking distance, is also discussed in this section.

• Applying the CE method. The aforementioned factors are all important for creating each solution in the population. In each iteration, after the population is created, the CE method is applied to optimise the total passenger walking distance. The first iteration in each execution of the optimisation process starts with the actions performed only once, i.e. the assessment of the state of the airport at the time of the execution. However, every iteration after that, until the optimal solution in that execution of the optimisation process has been found, starts with the section of creating the population. This can be done since the current state of the airport has already been assessed and recorded and can be referred back to without having to repeat the assessment.

Figure 11.2 is a schematic representation of the different sections in the optimisation process. This figure will be used as a road map throughout the discussion of the optimisation process in this chapter. Finally, a comparison between this study and the two case studies discussed in chapter 10 is made.

11.1 Overview of the optimisation process

The first part (optimisation overview) of the schematic representation of the optimisation model, shown in Figure 11.2, is discussed in this section. The meta-heuristic was coded in Simio and is performed in real time. Thus, as the model is run and as delays occur, the optimisation using the CE method is repeated to find a new flight-to-gate assignment schedule since the delay could not be considered the previous time the process was performed. If an aircraft is delayed and the optimisation process is not executed again, the original flight-to-gate assignment
11.1 Overview of the optimisation process

<table>
<thead>
<tr>
<th>OPTIMISATION OVERVIEW</th>
<th>ACTIONS PERFORMED ONCE</th>
<th>CREATE POPULATION</th>
<th>APPLY THE CE METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQUENCY OF PROCESS</td>
<td>FIRST ARRIVING FIGHT</td>
<td>GATE STATES</td>
<td>RELEASE GATES</td>
</tr>
<tr>
<td></td>
<td>FIRST DEPARTING FIGHT</td>
<td></td>
<td>OCCUPY GATES</td>
</tr>
<tr>
<td></td>
<td>LAST FIGHT</td>
<td></td>
<td>CALCULATE PERFORMANCE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 11.2: Schematic representation of the optimisation process

schedule is still used and the delayed flight will cause the next flight that is assigned to that gate to be delayed, and the next flight will delay the flight after that and so on. In the end almost all the flights will be delayed. It is therefore important that the CE method is applied each time a delay occurs.

The problem addressed in this optimisation task is more complicated than the average metaheuristic problem, since flights must be generated throughout the day. In a typical metaheuristic problem, where a snapshot is taken, there would be $n$ gates and $n$ flights with $n \times n$ combinations of which flight is parked at which gate. In that case, each flight can only be assigned to one gate and each gate can only accommodate one flight. That is a static problem. However, in this problem, flights are generated over time. As each flight arrives at the airport it is assigned to a gate and when it leaves the airport, the gate that it was assigned to is again available for a next flight. Here, the problem cannot be solved by using integer programming, since every gate can and will be occupied by more than one flight each day. This is thus a dynamic problem. As stated in section 9.2, the previous applications of the CE method include static, but not dynamic simulation problems. In this study, however, the CE method is applied
11.2 Frequency of performing the optimisation process

to a dynamic simulation problem.

The optimisation process is performed in zero simulation time. Thus, the flight that fired the optimisation process and all the other actions in the airport simulation are paused while the new flight-to-gate assignment schedule is being calculated. The optimisation process is then performed for an ideal situation, where no flights are delayed for any reason. This is done since without the simulation, where passengers and aircraft are actually moving on paths, there is no way to know if an aircraft will be delayed. When the optimisation process is finished the simulation is continued. As soon as there is a deviation from the ideal circumstances, the metaheuristic optimisation is repeated.

11.2 Frequency of performing the optimisation process

The optimisation process, using the CE method, is performed in each of the following cases:

- for the first flight
- for every 25th flight after the previous flight for which it was performed
11.2 Frequency of performing the optimisation process

- each time an aircraft is delayed
- each time the gate for which an arriving aircraft had to wait becomes available

If the arriving aircraft has a priority equal to the value of the priority of the flight that started the execution of the previous optimisation process plus 25, it is time to repeat the process. Each time the optimisation process is executed, i.e. the CE method is applied, it is executed for the flight currently arriving at the airport as well as the 50 flights scheduled to arrive after that.

If the scheduled arrival time of the aircraft is less than the actual arrival time, i.e. if the aircraft arrives late, then the optimisation process is also performed. However, in this study, the assumption is made that the only delays at the airport are caused by the airport itself. For example, because of an aircraft not being able to leave a gate at the scheduled departure time of the flight and before the next flight has to occupy that gate. Then the second aircraft is delayed. No arriving aircraft is delayed because of factors outside the control of the airport. This assumption can be made since the same circumstances are used in every model and layout, which means they can be compared.

When the aircraft arrives, it already has a gate to which it was assigned in the last execution of the optimisation process, since this process assigns the current arriving flight as well as the 50 flights arriving after that to gates. This is different from the models without the metaheuristic optimisation where the flights are only assigned to gates once they arrive at the airport (Rule 1 and Rule 2). If the gate to which the aircraft was assigned in the previous execution of the optimisation process is not available, i.e. it is still occupied by the previous aircraft assigned to it, the aircraft cannot go to that gate and must thus be assigned to a new gate. This means that the previous aircraft did not leave the gate at the scheduled departure time, because of a delay.

Here, the optimisation process is executed again, and now the flight cannot park at the gate to which it was originally assigned, since it is known that even though this gate is supposed to be available, it is not. If this is the case, the model must keep track of all the gates to which this first flight cannot be assigned in this run of the optimisation process. This is done by using a variable $\text{WrongGate}(x)$.
11.2 Frequency of performing the optimisation process

where $x$ is the gate number. If the value of $\text{WrongGate}(x)$ equals one, the current arriving aircraft cannot be assigned to it, and if the value is zero, the aircraft can be assigned to it. When the aircraft is finally assigned to a gate that is available and to which it can start taxiing right away, the variable $\text{WrongGate}(x)$ for $x$ equals one to the number of gates at the airport (122) is again set to zero. This must be done so that when the next flight arrives for which the optimisation process must be performed, it can be assigned to all the available gates. If this value is not set to zero, then the first arriving flight in the next execution of the optimisation process will not be allowed to park at this gate even though it may have become available in the mean time.

When the optimisation process is performed due to the case described above, i.e. a gate that is supposed to be available is not available, there is no way for the model to know how long the aircraft currently occupying the gate will be delayed. Thus, the gate is left out of the equation for a selected duration (15 minutes). When this duration has expired, the gate can be used again. However, the gate may become available before these 15 minutes are over. When the optimisation process was performed, it was assumed that the gate would be occupied for 15 more minutes, thus all the flights were assigned to gates while taking that into consideration. Now, after the optimisation process was performed, the gate becomes available. The previous calculated flight-to-gate schedule is not as good as it could have been when the correct time for the gate to be used again, was used. Thus, the optimisation process is executed again with the gate being available.
At the start of creating each random feasible solution in the population, the current state of the airport must be used. This state is changed when the state of a gate in the main simulation changes from available to occupied or vice versa. All the actions described in this section are performed only once in each execution of the optimisation process. These actions are not performed for every population member/solution or every iteration in the optimisation. They are performed in preparation for the process of creating the population of feasible solutions. The current state of the airport is assessed and recorded. The recorded values of the applicable variables are stored in variables that are referred to as “original”.

After each solution in the population has been generated, the variables representing the current state of the airport in the main simulation must be reset to the original values. The variables that must be reset include the arriving and departing flights with which the flight-to-gate assignment process must start in each solution, the flight with which to end the assignment in each solution and whether each gate at the airport is available or occupied at the start of the optimisation. For example, the state of a gate (available or occupied) will change over time while creating each solution in the population. However, after that solution is created the state of every gate is again set to the original state, i.e.
11.3 Actions performed only once

how it actually is in the main simulation model that is put on hold while the optimisation process is performed. This is done in order to create the next solution in the population while considering the correct circumstances at the airport.

11.3.1 Selecting the first arriving flight in the run

When the metaheuristic optimisation is started, the flight-to-gate schedule must be calculated for the first arriving flight that have not yet parked at a gate, up to 50 flights after that. The first arriving flight not yet at a gate can be the first flight in the simulation run, which is the case only when the optimisation process is executed for the first time. The first flight in the run can also be the flight arriving 25 flights after the flight for which the optimisation process was executed the last time. Furthermore, this first flight in the run can be a flight that cannot park at the gate to which it was assigned due to the fact that the gate is still occupied by the flight previously assigned to it. Finally, the first flight in the run can be the first flight not yet at a gate, after the gate that was still occupied when it was supposed to be available for use by another aircraft, becomes available.

If the first flight in the optimisation process is the first flight of the simulation run, the variable representing the first flight that must be assigned to a gate is
11.3 Actions performed only once

assigned the value one. If it is not the first flight, it must be determined whether the execution of the metaheuristic optimisation was fired by an arriving flight, or by a departing flight leaving a gate that was supposed to be available and thus caused the execution of the optimisation process. If the optimisation process was activated by an arriving flight, the flight with which to start the assignment schedule calculation is this arriving flight. If the optimisation process was caused by a departing flight, the arriving flight with which to start the calculation of the flight-to-gate assignment schedule is the first flight having a scheduled arrival time larger than or equal to the current simulation time.

![Diagram](image.png)

**Figure 11.3:** The first arriving flight to consider in the optimisation process

Figure 11.3 illustrates the first arriving flight with which the optimisation process starts. This is the first flight arriving after the simulation time at which the execution of optimisation process is started.
11.3.2 Selecting the first departing flight in the run

When starting the execution of the optimisation process, the flight with the earliest departure time, of which the departure time is bigger than or equal to the current simulation time, is the first flight to take into consideration in the optimisation process for departing. If it is the first execution of the optimisation process, this flight is the one with the overall earliest departure time, since no flights have arrived yet. If it is not the first execution, the departing flight with which to start is the first flight having a scheduled departure time plus a buffer time of bigger than or equal to the current simulation time. This buffer time is five minutes and is included to account for the time it takes the aircraft to taxi away from the gate and to allow a little time for delay so that the optimisation process will not need to be executed too often like when an aircraft leaves the gate at which it was parked only a few seconds late.

In Figure 11.4, the first departing flight that will be considered in the optimisation process is pointed out. This is the first flight departing after the simulation time at which the execution of the optimisation process is started.
11.3 Actions performed only once

Figure 11.4: The first departing flight to consider in the optimisation process

11.3.3 Assigning the last flight in the run

As stated before, every time the optimisation process is performed, flights are assigned to gates from the first flight that has not yet parked at a gate to 50 flights after that. The optimisation process thus plans 50 flights ahead. If no delays occur, this is repeated when the 25th flight after the previous first flight of the run arrives. Thus, the process can be described as shown in Figure 11.5.

The process shown in the figure is for an ideal situation, with no delays. If, for example the 32nd flight is delayed due to the fact that the gate to which it was assigned is still occupied by the previous aircraft, the previous calculated
11.3 Actions performed only once

Figure 11.5: Execution of the optimisation process

flight-to-gate assignment process will not be optimal anymore. The optimisation process must thus be repeated to find the new near-optimal solution. Then the process will be as depicted in Figure 11.6.

Figure 11.6: Execution of the optimisation process with delays

After the first arriving flight in the flight-to-gate schedule has been determined, the flight at which the model should stop assigning gates, i.e. the last flight, can be calculated by adding 50 to the value of the flight number of the first flight in the optimisation process. If the value of the number of this last flight is bigger than the total number of flights in the model, then the last flight in the model becomes the last flight in the calculation of the flight-to-gate assignment schedule. The reason for planning 50 flights ahead is because each flight has an influence on a number of flights arriving after it. It was observed that when this number of flights is set equal to 50, the time it takes to execute the optimisation process is feasible and the results are satisfactory.
11.3 Actions performed only once

11.3.4 Test for available gates

Every gate in the model is assessed. If the gate is currently occupied in the simulation model, it is recorded as unavailable in the optimisation process. However, if the departure time of the aircraft currently at that gate plus the buffer time of five minutes has been reached, i.e. the aircraft is not supposed to be at the gate anymore even though it is, the gate is recorded as available in the optimisation process. This is done since another variable keeps track of those gates that are supposed to be available, but are occupied. Also, it is necessary that these gates appear to be available, since the aircraft by which they are occupied were supposed to have left. In other words, the departure times of these aircraft have already been reached and the departing flight that the optimisation process will start with has a departure time greater than the departure times of those other aircraft. Thus, if the gates that they are occupying are not recorded as available, they will never become available in this execution of the optimisation process since the time in the optimisation process will already have passed the departure times of those flights.
11.4 Creating the population

In each execution of the optimisation process using the CE method, as stated before, a number of iterations is performed in order for the solutions to converge to the optimum. In each iteration, a population of feasible members is created. Each of these members is a solution as to which flights are assigned to which gates. Each solution has an objective which represents the total walking distance of all arriving and departing passengers in that solution. This objective will be minimised in the single objective optimisation problem.

The population in the optimisation process is represented by a matrix. The number of rows in the matrix is equal to the population size and the number of columns is equal to the number of flights considered in each run, i.e. 50. The population size in this study is 100. Thus, 100 feasible solutions are created in each iteration of the CE method. The value of Population(13, 97) will be the gate to which flight 97 is assigned in the 13th population member/solution.

When generating each solution, the following is considered:

- A flight may not be assigned to a gate before the flight has arrived.
- The gate to which a flight was assigned may not be available to another flight before this flight has departed.
11.4 Creating the population

- Each flight may only be assigned to one gate.
- No gate may be occupied by more than one flight at the same time.
- A large aircraft may not be assigned to a small gate.

Every flight arrives only once at the airport and can thus be assigned to only one gate. However, the gates are occupied and released over time as the aircraft arrive and depart. Many flights are thus assigned to each gate, but there may never be two or more flights assigned to a specific gate at the same time. A flight can thus only be assigned to a gate if the arrival time of the flight is greater than the departure time of the aircraft by which the gate was previously occupied.

The optimisation process must thus keep track of the simulated time while it is being executed. However, during the execution of the optimisation process, the simulation time in the model is put on hold since the model and all the processes in the model are paused while the new flight-to-gate assignment schedule is calculated so that after this calculation, the model can carry on from the simulation time at which it was stopped to execute the optimisation process. The simulation time can thus not be used to determine when an aircraft arrives or departs in the optimisation process. The process is thus a simulation inside the simulation model that is performed in zero time of the main simulation model, but that uses its own time. A variable called "timer" is used in the optimisation process to keep track of the time. At the beginning of the creation of each population member/solution, the value of the timer is set to the actual simulation time in the main model since that is the time from which the new schedule must be calculated. Also, the first arriving and departing flights to consider in the calculation of the solution and the state of each gate (occupied/available) are reset to the original values (which represent the current state in the simulation model).

In the following two sections, the generation of each solution in the population is discussed. Flights arrive and depart over time and the state of each gate thus changes from occupied to available and vice versa over time. Flights cannot be assigned to gates that are occupied. The state of each gate must therefore be known in order to decide whether a flight may be assigned to it. The processes of releasing and occupying the gates are described in sections 11.4.1 and 11.4.2.
11.4 Creating the population

11.4.1 Releasing the gates when the flights depart

The first departing flight for the flight-to-gate assignment process is considered. If the departure time of this flight plus the buffer time of five minutes is larger than or equal to the current value of the timer, then the gate is released. The value of the departing flight that is being considered is incremented and the next flight is considered. The process described above is repeated until the departure time of the flight in consideration plus the buffer time is less than the value of the timer, i.e. until the first flight of which the departure time has not been reached is considered.

If the airport is very busy it may happen that a flight is still waiting to be assigned to a gate in the optimisation process when the departure time of that flight is reached. In other words, it is already time for the aircraft to release the gate while it has not even been assigned to a gate. This will not happen very often since an airport cannot be forced to handle a schedule that is too busy and that will result in delaying the entire air system. However, sometimes this exceptional situation may occur and must thus be accounted for.

When the aircraft assigned to a gate is assessed to determine whether the departure time of this aircraft has been reached, the condition of whether the aircraft has been assigned to a gate yet must also be satisfied. If this condition
11.4 Creating the population

is not satisfied, the gate cannot be released since the flight of which it must be released has not even occupied the gate yet. Thus, the value of $\text{Population}(y, i)$ where $y$ is the population member/solution currently being calculated and $i$ is the flight of which the gate must be released, is zero since flight $i$ has not yet been assigned to a gate in the $y^{th}$ solution. It will thus not be known in the optimisation process which gate to release. In this case the number of this flight is stored to be referred to later in order to test if it has been assigned to a gate. This number is stored in the variable $\text{MNotAssigned}(dcounter)$ where $dcounter$ is the number of flights that are already supposed to have departed but that have not yet been assigned to gates and that must thus be referred to later. Every time such a case occurs, $dcounter$ is incremented and the flight number in this case is stored in that place. These flights, of which the scheduled departure time have been reached but that have not yet been assigned to gates, will need to be evaluated each time before a new departing flight is tested.

Whenever a new departing flight is tested to determine whether it has departed and whether the gate which it is supposed to occupy can be released, the condition of whether the value of $dcounter$ is bigger than one is tested. If this condition is satisfied, it means that there are flights of which the departure times have been reached, but that were not yet assigned to gates the last time they were evaluated. Since that last time, there may have been gates that became available to which they could have been assigned. It is thus necessary to test these flights again to determine whether the gate to which they were assigned can be released. The first of these flights is tested. If this flight has still not been assigned to a gate, the next flight in this group of flights is tested until all the flights that were previously waiting to be assigned to gates have been tested.

If the flight that is being evaluated has been assigned to a gate, the gate is released. The condition of whether the flight had to wait to be assigned to a gate after the departure time had been reached is tested and in this case it is true. Then, if this flight was the first flight in the group of flights that had to wait to be assigned to a gate, the variable referring to the first flight in this group that must be tested is incremented so it will start testing at the next flight when the flights are evaluated again. This is done so that the flight that was just considered is not considered again, thus preventing the gate to which it was assigned to be released.
11.4 Creating the population

twice for one flight. After this, the next flight is evaluated. When all the flights in this group of flights have been evaluated, the departure time of the next flight, that has not yet been evaluated and that is not part of this group, is tested to determine whether it is bigger than the current value of the timer. If it is not bigger or equal to that value, the gate at which this aircraft is parked cannot be released yet and the optimisation process continues to test the arriving flights.

11.4.2 Occupying the gates when the flights arrive

If the value of the timer in the optimisation process is bigger than or equal to the arrival time of the first arriving flight considered in this calculation of the flight-to-gate assignment schedule, it means that this flight would already have arrived and must thus be assigned to a gate. A random value between zero and one is then selected.

The probabilities that flight \( i \) will park at each gate are changed when the CE method is applied in order to make the solutions converge to the optimal solution. These probabilities, that were previously calculated, must be reviewed since some gates that were previously occupied may now be available or vice versa. When the probability that flight \( i \) will be assigned to gate \( j \) is set to zero, due to the fact that gate \( j \) is currently occupied, the actual probability is stored in another
variable. This is done so that if the gate becomes available again, the original probability can be used as it should be.

Each gate is evaluated. If the gate is available and the flight that was previously assigned to it is not delayed, it should be available for the flight in question if it is a suitable gate for the aircraft. If the variable that stores the probability temporarily so that the probability can be recalled is equal to zero, it means that the probability was not previously changed and nothing needs to be done. If the value of the temporary variable is not zero, it means that the actual probability is being stored and must thus be recalled. As soon as this original probability has been recalled, the value of the temporary variable is set to zero.

If the gate that is being evaluated appears to be available since the flight by which it was previously occupied was supposed to have left by this time, but that flight has in fact not left, then the probability that flight $i$ can be assigned to that gate must be set to zero while the actual probability must be temporarily stored. This is the same for instances where the gate does not appear to be available due to the fact that the departure time of the flight by which it is currently occupied has not yet been reached.

If flight $i$ is the first arriving flight considered in this execution of the optimisation process, in other words if it is the flight currently arriving in the main simulation model which caused the execution since the gate to which it was assigned is not available, it cannot be assigned to this gate again. It is not necessarily just one gate that cannot be used for this flight due to this reason, but all the gates for which the value of the variable $\text{WrongGate}(x)$, where $x$ is the number of the gate, equals one as explained in section 11.2. The probabilities of this first flight being assigned to any of the gates, $x$, of which the value of $\text{WrongGate}(x)$ is one, is set to zero and the original probabilities are stored temporarily.

If gate $j$ is a small gate and flight $i$ is a large aircraft then the probability that flight $i$ can park at gate $j$ equals zero. An empirical distribution of these probabilities is used to pick a random suitable gate by using the random number between zero and one that was selected.

Figure 11.7 represents this empirical distribution for a flight (flight $i$) at an airport with seven gates. The probability that flight $i$ can park at gate 1 is 0.1 and the probability for gate 2 is 0.2 etc. In this example, gates 3 and 6 are either
11.4 Creating the population

![Figure 11.7: The empirical distribution of the probability of flight \( i \) parking at gate \( j \)](image)

not suitable for flight \( i \) or they are unavailable. No matter what random number between zero and one is selected, flight \( i \) will not be assigned to gate 3 since the probability that flight \( i \) will park at gate 3 equals zero. If the selected random number is between 0.1 and 0.3, gate 2 will be used for flight \( i \). Also, as can be seen from the figure, gate 7 has the highest probability to be selected (0.4) and the chance that the selected random number will result in choosing gate 7 is bigger than for the other gates.

In the first iteration of the metaheuristic, the probability that flight \( i \) will be assigned to each gate is the same for all the suitable gates. In the later iterations, after the cross-entropy method has been performed, these probabilities will have changed and those gates that will result in an overall smaller passenger walking distance will have larger probabilities to be selected than the others. After the random number is selected and it has been decided to which gate to assign flight \( i \), the flight is assigned to it and the status of the gate is changed to occupied.

If the sum of the probabilities that flight \( i \) can park at any gate \( (j) \) equals zero, thus:
11.4 Creating the population

\[
\sum_{j=1}^{n} \text{Prob}(j,i) = 0, \quad (11.1)
\]

it means that no suitable gates are available for flight \(i\) to park at. Here \(n\) denotes the number of gates, which is 122 in this problem. Flight \(i\) can thus not be assigned to a gate and is delayed until a gate becomes available. It is in this case that the departure time of a flight may be reached before the flight has been assigned to a gate. However, the next arriving flight must be evaluated since there may be a suitable gate available for this flight due to the fact that this flight is of a different size than the one for which no gates are available. Thus, the value of the variable “timer” will be incremented and those flights not yet assigned, but of which the arrival times have been reached, must be considered later and their priorities are therefore stored in another variable. This is done in a similar way as when the departure time of a flight has been reached and a gate must be released, while the flight has not yet been assigned to a gate, as explained in section 11.4.1. Again, every time before a new arriving flight is considered to be assigned to a gate, these flights that have already arrived, but that still need to be assigned to gates, are evaluated to determine if a suitable gate has become available in the meantime.

If a flight had to wait to be assigned to a gate in the final solution that will be used by the main simulation model after the optimisation process is complete, and the flight arrives at the airport, the gate to which it was assigned will not be available. However, since the aircraft is supposed to wait, the optimisation process is not executed again like when a flight that is not supposed to wait arrives while the gate to which it was assigned is not available. Only after the calculated waiting time for the flight has expired and the gate to which it was assigned is still occupied, the optimisation process is performed again.

When all 50 flights that are supposed to be assigned to gates in the optimisation process have been assigned to gates, all the values are reset to again represent the current state in the simulation. After this, the next solution of the population is calculated until all 100 members in the population have been completed.
11.4 Creating the population

11.4.3 Calculating the performance of each solution

When a flight is assigned to a gate in a solution of the metaheuristic optimisation, the distance from the terminal building to that gate is determined. This value is then multiplied by the number of arriving passengers on that flight plus the number of departing passengers on the next flight of that aircraft. The walking distances for passengers on each flight in the solution are then summed to calculate the total walking distance of passengers in that solution. This value of each solution will later be evaluated to determine which solutions are the best and should be used to adjust the probabilities of specific flights being assigned to specific gates.
An example of a population to which the CE method can be applied is shown in Table 11.1. In this example, there are ten flights that can each be assigned to any one of five gates (numbered 1 to 5). The example represents a population consisting of ten members/solutions. In this example, flight 6 is assigned to gate 4 in the first solution in the population, to gate 2 in the second solution etc. When referring to Algorithm 4 in section 9.2.2, the sample \( X_1, \ldots, X_N \) generated from the density \( h(\cdot; \hat{v}_{t-1}) \), is represented by the solutions in this table.

The variable \( \text{MProb}(j,i) \) represents the probability that flight \( i \) will park at gate \( j \). Whenever it is impossible for flight \( i \) to park at gate \( j \), for example when flight \( i \) is a large aircraft and gate \( j \) is a small gate, the probability equals zero. In the first iteration of the metaheuristic optimisation, all probabilities for flight \( i \) are the same for any suitable gate. These probabilities change as the cross-entropy method is applied. Table 11.2 is an example of the probabilities that each flight can be assigned to each gate. This example represents ten flights that can be assigned to any one of five gates. In this example the probability of flight 6 being assigned to gate 3 is 0.1 and the probability of flight 6 being assigned to gate 5 is 0.7 etc. These probabilities represent the density, \( h(\cdot; \hat{v}_{t-1}) \), from which the solutions in the sample are generated in each iteration.
11.5 The application of the CE method

Table 11.1: An example of a population of feasible solutions

<table>
<thead>
<tr>
<th>Solution number</th>
<th>Flight number</th>
<th>Walking distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3 2 5 1 3 4 3 5 2 1</td>
<td>645 000.00</td>
</tr>
<tr>
<td>2</td>
<td>2 3 1 5 4 2 3 1 2 4</td>
<td>639 521.00</td>
</tr>
<tr>
<td>3</td>
<td>5 3 4 2 3 4 5 1 2 4</td>
<td>665 741.00</td>
</tr>
<tr>
<td>4</td>
<td>1 3 4 2 1 5 3 2 4 5</td>
<td>628 951.00</td>
</tr>
<tr>
<td>5</td>
<td>1 3 5 2 4 1 3 2 1 5</td>
<td>615 426.00</td>
</tr>
<tr>
<td>6</td>
<td>1 2 1 4 3 5 2 1 3 4</td>
<td>654 897.00</td>
</tr>
<tr>
<td>7</td>
<td>4 3 5 2 4 1 4 2 3 1</td>
<td>634 159.00</td>
</tr>
<tr>
<td>8</td>
<td>3 5 2 4 3 1 5 3 2 4</td>
<td>614 598.00</td>
</tr>
<tr>
<td>9</td>
<td>5 2 3 5 2 3 1 2 4 2</td>
<td>632 541.00</td>
</tr>
<tr>
<td>10</td>
<td>4 3 5 1 3 2 5 4 3 1</td>
<td>654 123.00</td>
</tr>
</tbody>
</table>

Table 11.2: Probabilities of assigning flights to gates

<table>
<thead>
<tr>
<th>Gate number</th>
<th>Flight number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5 6 7 8 9 10</td>
</tr>
<tr>
<td>1</td>
<td>0.3 0.1 0.0 0.1 0.0 0.0 0.4 0.1 0.1 0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.1 0.3 0.2 0.4 0.2 0.1 0.3 0.0 0.5 0.6</td>
</tr>
<tr>
<td>3</td>
<td>0.2 0.0 0.4 0.1 0.3 0.1 0.1 0.2 0.1 0.2</td>
</tr>
<tr>
<td>4</td>
<td>0.4 0.5 0.1 0.2 0.1 0.1 0.0 0.3 0.1 0.1</td>
</tr>
<tr>
<td>5</td>
<td>0.0 0.1 0.3 0.2 0.4 0.7 0.2 0.4 0.2 0.1</td>
</tr>
</tbody>
</table>

In this study, however, a population of 100 members is created each time the optimisation process is repeated. When all 100 feasible solutions in the population have been created, they must be sorted according to the overall passenger walking distance, since the objective of this process is to minimise this distance. The population is thus sorted so that the solutions with the smallest total walking distance are at the bottom of the population. The 20% of the solutions with the lowest total walking distances are then used to calculate the probabilities that each flight will park at each gate, thus \( \text{MProb}(j, i) \) where \( j \) is the gate number and \( i \) is the flight number. Table 11.3 shows the sorted population of the above discussed example, and the solutions that will be selected in calculating the new
11.5 The application of the CE method

probabilities. These solutions are the 20% of the solutions with the smallest total passenger walking distance. In the case of the example, since the population consists of ten solutions, two solutions will be selected for this purpose. From Table 11.3 it can be observed that the eighth solution has the smallest total passenger walking distance and that solution 3 has the largest total passenger walking distance.

Table 11.3: Sorted population

<table>
<thead>
<tr>
<th>Solution number</th>
<th>Flight number</th>
<th>Walking distance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5 3 4 2 3 4 5 1 2 4</td>
<td>665 741.00</td>
</tr>
<tr>
<td>6</td>
<td>1 2 1 4 3 5 2 1 3 4</td>
<td>654 897.00</td>
</tr>
<tr>
<td>10</td>
<td>4 3 5 1 3 2 5 4 3 1</td>
<td>654 123.00</td>
</tr>
<tr>
<td>1</td>
<td>3 2 5 1 3 4 3 5 2 1</td>
<td>645 000.00</td>
</tr>
<tr>
<td>2</td>
<td>2 3 1 5 4 2 3 1 2 4</td>
<td>639 521.00</td>
</tr>
<tr>
<td>7</td>
<td>4 3 5 2 4 1 4 2 3 1</td>
<td>634 159.00</td>
</tr>
<tr>
<td>9</td>
<td>5 2 3 5 2 3 1 2 4 2</td>
<td>632 541.00</td>
</tr>
<tr>
<td>4</td>
<td>1 3 4 2 1 5 3 2 4 5</td>
<td>628 951.00</td>
</tr>
<tr>
<td>5</td>
<td>1 3 5 2 4 1 3 2 1 5</td>
<td>615 426.00</td>
</tr>
<tr>
<td>8</td>
<td>3 5 2 4 3 1 5 3 2 4</td>
<td>614 598.00</td>
</tr>
</tbody>
</table>

For the populations which each consists of 100 solutions, created in the optimisation process, the 20 solutions with the smallest total passenger walking distance are considered in the calculation of the new probabilities. The number of times flight $i$ parked at gate $j$ in these 20 solutions is counted for each flight considered in the run. For example, if flight 15 parked at gate 3 in five out of the 20 solutions, then the new calculated probability $MProbCalculated(3, 15)$ will be equal to 0.25. These new calculated probabilities are then used to adjust the existing probabilities. Equation 9.16 in chapter 9 is used for this purpose as shown below:

$$MProbNew(j, i) = 0.8 \times MProbCalculated(j, i) + 0.2 \times MProb(j, i)$$ (11.2)
11.6 The generic model of the optimisation process using the CE method

The values of $M\text{ProbNew}(j, i)$ are then used to select gates to assign the flights to in the following iteration of the metaheuristic optimisation, i.e. when the next population is created. If gate $j$ is a small gate and the aircraft of flight $i$ is a large aircraft, then the value of $M\text{Prob}(j, i)$ would be zero from the start. Also, flight $i$ would never be assigned to gate $j$ in any of the solutions of the population which means the value of $M\text{ProbCalculated}(j, i)$ will also be zero. This will result in the value of $M\text{ProbNew}(j, i)$ to also be zero as it should be.

After a specified number of iterations has been performed in the optimisation process, or when the performances of the solutions in the population of the last iteration are satisfactory, the optimisation process is terminated. The best solution in the last iteration of the process is then used in the main simulation model for assigning flights to gates. The main simulation is then continued until it is time for the next execution of the optimisation process.

11.6 The generic model of the optimisation process using the CE method

The model developed to reduce the passenger walking distances at the airport, by improving the flight-to-gate assignment process using the CE method, is a generic model and can be applied to any airport model of this nature. This model includes the following aspects as described in this chapter:

- applying the optimisation process in real-time, thus for every $n^{th}$ flight and each time a delay occurs
- assessing the current state of the airport before each execution of the optimisation process
- creating the population in each iteration of the optimisation process
- applying the CE method and adjusting the probabilities, of each flight being assigned to each gate, after each iteration

Variables and tables that need to be adjusted to represent the specific airport to which the model is applied include:
11.7 This study versus the case studies

- the flight schedule
- number of gates at the airport
- number of flights simulated
- the distances from each gate to the terminal building
- the distance from each gate to the runways
- the number of iterations to be performed each time the optimisation process is executed
- the number of flights to consider each time the optimisation process is performed (50 flights were used in this study)
- the frequency of performing the optimisation process if there are no delays (25 flights were used in this study)
- the population size in each iteration of the optimisation process

This generic model of the optimisation process was applied to each of the four apron layout designs to determine the effect on the passenger walking distances when using the CE method.

11.7 This study versus the case studies

In the first case study, by Yan et al. discussed in chapter 10, delays at the airport are considered and the optimisation process is performed in real time, but a buffer period of 30 minutes is used to account for the delays. Furthermore, when a delay of longer than that buffer period occurs, the optimisation process is not performed again, instead the aircraft is just assigned to another gate without considering different solutions and selecting the best one. In this study, however, the buffer period is only five minutes, thus the delays are considered much more accurately. Also, when a delay of more than the buffer period occurs, the optimisation process is repeated to find the new solution that will ensure a smaller total walking distance.
In the second case study, by Drexl and Nikulin, multi-objective optimisation is used with the objectives being to minimise the number of aircraft parked on the apron, to minimise the total passenger walking distance and to maximise the preference for assigning specific flights to specific gates. Furthermore, in that study the optimisation process is not performed in real time. However, in this study, there is no space for aircraft to park on the apron which means all aircraft must be assigned to gates, and there is no preference for specific flights to be assigned to specific gates.

### 11.8 Concluding remarks chapter 11

In order to reduce the passenger walking distance in the airport, a generic optimisation model using single objective optimisation, specifically the cross-entropy method, was developed and applied to each of the simulation models. This was focused on the flight-to-gate assignment process used in Rule 3. Previous application of the cross-entropy method includes static simulation problems and not dynamic simulations. However, in this study the optimisation process using the CE method was performed in real time to calculate a new flight-to-gate assignment schedule each time a delay occurred at the airport. The CE method was thus applied as a real-time optimiser, and it is proposed in this research that it can be used as such in real-world decision-support systems. This concludes the problem solving phase of the study. The results of the different scenarios that were simulated, including those using Rule 1, Rule 2 and Rule 3 (the meta-heuristic optimisation) for assigning flights to gates will be provided in chapter 12.
Chapter 12

Data and results

There are two models of each airport design. The first model is run with the wind coming from the one side of the runway, and second with the wind coming from the other side. The wind direction has a large influence on the direction in which the aircraft arrive and depart. For both of these actions, the wind has to be from the front in order to provide aircraft lift (Londoño, 2010). Thus, the wind will determine from which direction the aircraft will approach the apron. This will have an effect on the distance the aircraft travel on the taxiways as well as on the time each aircraft spends in the system. It is necessary to have a model
of each in order to quantify the effect of the wind direction. Each of these models was designed based on each of the three rules used to assign flights to gates.

12.1 The data used in the models

The data used in the simulation models is based on that of OR Tambo International Airport of 9 - 11 October 2007. According to Van Zyl (2011), this was a very busy time at the airport. However, the data was scaled to find the busiest schedule the new airport can handle. A scaling programme was written to adjust the data. The logic in this programme is as follows:

- The scaling factor is selected by the user.

- The number of aircraft arriving each hour in the OR Tambo schedule is counted and then multiplied by the scaling factor to find the new calculated number of flights for that hour.

- Flights in the existing OR Tambo schedule are then randomly selected to be used in the new schedule until there are enough flights according to the calculated number for the new airport. A random number between -15 and 15 is then added to the arrival and departure times of each of these selected flights to prevent them from arriving at the exact same time.

- The flights for the next hour is then calculated.

Different scaling factors were tested. The resulting data was then used in the models in order to find a suitable schedule for the airport designs.

12.2 The results of the simulation models

As explained before, different scenarios were simulated in this study. For each of the four airport layouts, two models were simulated to account for the wind direction. Also, each of these was run based on the three rules for assigning flights to gates. In the first run, Rule 1 was used in selecting a gate for an aircraft, in the second run Rule 2 was used and in the third run Rule 3 was
12.2 The results of the simulation models

used, i.e. metaheuristic optimisation using the CE method. Thus, a total of 24 scenarios were simulated.

Since at the start of each run the airport is empty, it is important to include a warm-up period in order to get a realistic value for the average number of aircraft present at the airport. It was observed that after approximately 12 simulation hours, the airport is in steady state. Thus, a warm-up period of 12 hours was included.

12.2.1 The statistics used to compare the scenarios

Table 12.1 shows all the statistics used in the models as well as the type of statistics.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Statistic type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arriving passengers’ total time in system</td>
<td>Output</td>
</tr>
<tr>
<td>Arriving passengers’ total walking distance</td>
<td>Output</td>
</tr>
<tr>
<td>Number of aircraft in the system</td>
<td>State</td>
</tr>
<tr>
<td>Departing passengers’ total time in system</td>
<td>Output</td>
</tr>
<tr>
<td>Departing passengers’ total total walking distance</td>
<td>Output</td>
</tr>
<tr>
<td>Number of departing aircraft</td>
<td>Output</td>
</tr>
<tr>
<td>Time in system of each aircraft</td>
<td>Tally</td>
</tr>
<tr>
<td>Total aircraft circling time</td>
<td>Output</td>
</tr>
<tr>
<td>Total aircraft delayed time</td>
<td>Output</td>
</tr>
<tr>
<td>Total departing passengers’ overtime</td>
<td>Output</td>
</tr>
<tr>
<td>Aircraft total distance travelled on taxiways</td>
<td>Output</td>
</tr>
<tr>
<td>Total number of arriving passengers</td>
<td>Output</td>
</tr>
<tr>
<td>Total number of departing passengers</td>
<td>Output</td>
</tr>
<tr>
<td>Total passenger circling time</td>
<td>Output</td>
</tr>
<tr>
<td>Aircraft time spent taxiing</td>
<td>Output</td>
</tr>
</tbody>
</table>

These statistics were used to compare the different scenarios. The results of each scenario, based on these statistics, are presented in section 12.2.2.
12.2 The results of the simulation models

12.2.2 Presentation of the results

The results for the four airport designs are shown in the following tables. For each of the designs, the results for both models with regard to the wind direction are shown. The first set of results for each wind direction is for the model using Rule 1 for assigning flights to gates, the second is for the model using Rule 2 and the third is for using Rule 3, i.e. metaheuristic optimisation.
12.2 The results of the simulation models

Table 12.2: Results obtained for Design 1, wind direction 1

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Wind direction 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rule 1</td>
</tr>
<tr>
<td>Arriving passengers’ total time in system (hours)</td>
<td>57 900.23</td>
</tr>
<tr>
<td>Arriving passengers’ total walking distance (m)</td>
<td>119 248 485.00</td>
</tr>
<tr>
<td>Number of aircraft in the system: Average</td>
<td>75.11</td>
</tr>
<tr>
<td>Number of aircraft in the system: Final value</td>
<td>83.00</td>
</tr>
<tr>
<td>Number of aircraft in the system: Maximum</td>
<td>108.00</td>
</tr>
<tr>
<td>Departing passengers’ total time in system (hours)</td>
<td>81 664.26</td>
</tr>
<tr>
<td>Departing passengers’ total walking distance (m)</td>
<td>112 972 450.25</td>
</tr>
<tr>
<td>Number of departing aircraft</td>
<td>1 216.00</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Average</td>
<td>2.42</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Maximum</td>
<td>24.96</td>
</tr>
<tr>
<td>Total aircraft circling time (hours)</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft overtime (hours)</td>
<td>59.86</td>
</tr>
<tr>
<td>Total departing passengers’ overtime (hours)</td>
<td>11 279.34</td>
</tr>
<tr>
<td>Aircraft total distance travelled on taxiways (m)</td>
<td>2 780 062.00</td>
</tr>
<tr>
<td>Total number of arriving passengers</td>
<td>135 966.00</td>
</tr>
<tr>
<td>Total number of departing passengers</td>
<td>129 196.00</td>
</tr>
<tr>
<td>Total passenger circling time (hours)</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft time spent taxiing (hours)</td>
<td>118.99</td>
</tr>
</tbody>
</table>
Table 12.3: Results obtained for Design 1, wind direction 2

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3 (CE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arriving passengers’ total time in system (hours)</td>
<td>57 101.81</td>
<td>54 623.17</td>
<td>52 943.82</td>
</tr>
<tr>
<td>Arriving passengers’ total walking distance (m)</td>
<td>119 146 289.75</td>
<td>110 203 694.00</td>
<td>102 683 559.75</td>
</tr>
<tr>
<td>Number of aircraft in the system: Average</td>
<td>75.22</td>
<td>75.28</td>
<td>75.39</td>
</tr>
<tr>
<td>Number of aircraft in the system: Final value</td>
<td>83.00</td>
<td>83.00</td>
<td>83.00</td>
</tr>
<tr>
<td>Number of aircraft in the system: Maximum</td>
<td>108.00</td>
<td>110.00</td>
<td>109.00</td>
</tr>
<tr>
<td>Departing passengers’ total time in system (hours)</td>
<td>82 455.66</td>
<td>82 833.38</td>
<td>83 147.85</td>
</tr>
<tr>
<td>Departing passengers’ total walking distance (m)</td>
<td>112 841 972.25</td>
<td>105 074 981.75</td>
<td>97 371 755.75</td>
</tr>
<tr>
<td>Number of departing aircraft</td>
<td>1 216.00</td>
<td>1 216.00</td>
<td>1 216.00</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Average</td>
<td>2.42</td>
<td>2.42</td>
<td>2.43</td>
</tr>
<tr>
<td>Total aircraft circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft overtime (hours)</td>
<td>59.83</td>
<td>56.69</td>
<td>68.90</td>
</tr>
<tr>
<td>Total departing passengers’ overtime (hours)</td>
<td>11 252.72</td>
<td>11 288.25</td>
<td>11 262.45</td>
</tr>
<tr>
<td>Aircraft total distance travelled on taxiways (m)</td>
<td>2 781 133.50</td>
<td>2 784 730.50</td>
<td>3 090 822.00</td>
</tr>
<tr>
<td>Total number of arriving passengers</td>
<td>135 966.00</td>
<td>136 003.00</td>
<td>136 099.00</td>
</tr>
<tr>
<td>Total number of departing passengers</td>
<td>129 196.00</td>
<td>129 196.00</td>
<td>129 196.00</td>
</tr>
<tr>
<td>Total passenger circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft time spent taxiing (hours)</td>
<td>119.12</td>
<td>121.08</td>
<td>119.50</td>
</tr>
</tbody>
</table>
Table 12.4: Results obtained for Design 2, wind direction 1

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3 (CE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arriving passengers’ total time in system (hours)</td>
<td>45 783.35</td>
<td>43 747.22</td>
<td>42 746.73</td>
</tr>
<tr>
<td>Arriving passengers’ total walking distance (m)</td>
<td>69 627 381.00</td>
<td>63 757 338.50</td>
<td>59 238 542.50</td>
</tr>
<tr>
<td>Number of aircraft in the system: Average</td>
<td>74.99</td>
<td>74.95</td>
<td>74.97</td>
</tr>
<tr>
<td>Number of aircraft in the system: Final value</td>
<td>83.00</td>
<td>83.00</td>
<td>83.00</td>
</tr>
<tr>
<td>Number of aircraft in the system: Maximum</td>
<td>109.00</td>
<td>109.00</td>
<td>109.00</td>
</tr>
<tr>
<td>Departing passengers’ total time in system (hours)</td>
<td>81 614.27</td>
<td>81 308.44</td>
<td>81 654.19</td>
</tr>
<tr>
<td>Departing passengers’ total walking distance (m)</td>
<td>65 922 998.00</td>
<td>60 812 574.00</td>
<td>56 169 166.00</td>
</tr>
<tr>
<td>Number of departing aircraft</td>
<td>1 216.00</td>
<td>1 216.00</td>
<td>1 216.00</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Average</td>
<td>2.41</td>
<td>2.41</td>
<td>2.41</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Maximum</td>
<td>24.97</td>
<td>24.96</td>
<td>24.97</td>
</tr>
<tr>
<td>Total aircraft circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft overtime (hours)</td>
<td>50.78</td>
<td>49.99</td>
<td>53.36</td>
</tr>
<tr>
<td>Total departing passengers’ overtime (hours)</td>
<td>10 969.83</td>
<td>10 949.80</td>
<td>11 003.16</td>
</tr>
<tr>
<td>Aircraft total distance travelled on taxiways (m)</td>
<td>2 684 949.75</td>
<td>2 384 603.75</td>
<td>2 635 407.25</td>
</tr>
<tr>
<td>Total number of arriving passengers</td>
<td>135 966.00</td>
<td>136 003.00</td>
<td>136 155.00</td>
</tr>
<tr>
<td>Total number of departing passengers</td>
<td>129 196.00</td>
<td>129 196.00</td>
<td>129 196.00</td>
</tr>
<tr>
<td>Total passenger circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft time spent taxiing (hours)</td>
<td>115.85</td>
<td>107.16</td>
<td>109.00</td>
</tr>
</tbody>
</table>
Table 12.5: Results obtained for Design 2, wind direction 2

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3 (CE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arriving passengers’ total time in system (hours)</td>
<td>46 299.29</td>
<td>44 642.35</td>
<td>44 006.55</td>
</tr>
<tr>
<td>Arriving passengers’ total walking distance (m)</td>
<td>69 948 933.00</td>
<td>63 717 958.50</td>
<td>59 752 150.50</td>
</tr>
<tr>
<td>Number of aircraft in the system: Average</td>
<td>74.94</td>
<td>74.77</td>
<td>74.92</td>
</tr>
<tr>
<td>Number of aircraft in the system: Final value</td>
<td>83.00</td>
<td>83.00</td>
<td>83.00</td>
</tr>
<tr>
<td>Number of aircraft in the system: Maximum</td>
<td>108.00</td>
<td>109.00</td>
<td>109.00</td>
</tr>
<tr>
<td>Departing passengers’ total time in system (hours)</td>
<td>81 183.16</td>
<td>80 504.46</td>
<td>80 461.54</td>
</tr>
<tr>
<td>Departing passengers’ total walking distance (m)</td>
<td>66 256 782.00</td>
<td>60 781 818.00</td>
<td>56 653 650.00</td>
</tr>
<tr>
<td>Number of departing aircraft</td>
<td>1 216.00</td>
<td>1 216.00</td>
<td>1 216.00</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Average</td>
<td>2.41</td>
<td>2.40</td>
<td>2.41</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Maximum</td>
<td>24.95</td>
<td>24.95</td>
<td>24.95</td>
</tr>
<tr>
<td>Total aircraft circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft overtime (hours)</td>
<td>50.79</td>
<td>49.90</td>
<td>53.26</td>
</tr>
<tr>
<td>Total departing passengers’ overtime (hours)</td>
<td>10 969.45</td>
<td>10 947.08</td>
<td>10 998.79</td>
</tr>
<tr>
<td>Aircraft total distance travelled on taxiways (m)</td>
<td>2 684 837.75</td>
<td>2 389 941.75</td>
<td>2 653 094.25</td>
</tr>
<tr>
<td>Total number of arriving passengers</td>
<td>135 966.00</td>
<td>136 003.00</td>
<td>136 003.00</td>
</tr>
<tr>
<td>Total number of departing passengers</td>
<td>129 196.00</td>
<td>129 196.00</td>
<td>129 196.00</td>
</tr>
<tr>
<td>Total passenger circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft time spent taxiing (hours)</td>
<td>115.83</td>
<td>107.37</td>
<td>108.43</td>
</tr>
</tbody>
</table>
### Table 12.6: Results obtained for Design 3, wind direction 1

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3 (CE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arriving passengers’ total time in system (hours)</td>
<td>51 958.73</td>
<td>50 661.40</td>
<td>48 949.99</td>
</tr>
<tr>
<td>Arriving passengers’ total walking distance (m)</td>
<td>87 495.77</td>
<td>81 424.36</td>
<td>80 968.17</td>
</tr>
<tr>
<td>Number of aircraft in the system: Average</td>
<td>75.01</td>
<td>74.89</td>
<td>75.16</td>
</tr>
<tr>
<td>Number of aircraft in the system: Final value</td>
<td>83.00</td>
<td>83.00</td>
<td>83.00</td>
</tr>
<tr>
<td>Number of aircraft in the system: Maximum</td>
<td>108.00</td>
<td>109.00</td>
<td>109.00</td>
</tr>
<tr>
<td>Departing passengers’ total time in system (hours)</td>
<td>87 936 241.00</td>
<td>81 175 570.00</td>
<td>71 206 592.25</td>
</tr>
<tr>
<td>Number of departing aircraft</td>
<td>1 216.00</td>
<td>1 216.00</td>
<td>1 216.00</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Average</td>
<td>2.41</td>
<td>2.41</td>
<td>2.42</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Maximum</td>
<td>24.96</td>
<td>24.95</td>
<td>24.95</td>
</tr>
<tr>
<td>Total aircraft circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft overtime (hours)</td>
<td>55.03</td>
<td>52.81</td>
<td>62.07</td>
</tr>
<tr>
<td>Total departing passengers’ overtime (hours)</td>
<td>11 147.53</td>
<td>11 148.75</td>
<td>11 152.60</td>
</tr>
<tr>
<td>Aircraft total distance travelled on taxiways (m)</td>
<td>2 761 689.50</td>
<td>2 779 735.00</td>
<td>3 073 281.50</td>
</tr>
<tr>
<td>Total number of arriving passengers</td>
<td>135 966.00</td>
<td>136 003.00</td>
<td>136 127.00</td>
</tr>
<tr>
<td>Total number of departing passengers</td>
<td>129 196.00</td>
<td>129 196.00</td>
<td>129 196.00</td>
</tr>
<tr>
<td>Total passenger circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft time spent taxiing (hours)</td>
<td>118.46</td>
<td>120.43</td>
<td>120.07</td>
</tr>
</tbody>
</table>
12.2 The results of the simulation models

<table>
<thead>
<tr>
<th>Design 3</th>
<th>Wind direction 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rule 1</td>
</tr>
<tr>
<td>Arriving passengers’ total time in system (hours)</td>
<td>51 003.09</td>
</tr>
<tr>
<td>Arriving passengers’ total walking distance (m)</td>
<td>92 867 601.75</td>
</tr>
<tr>
<td>Number of aircraft in the system: Average</td>
<td>75.14</td>
</tr>
<tr>
<td>Number of aircraft in the system: Final value</td>
<td>83.00</td>
</tr>
<tr>
<td>Number of aircraft in the system: Maximum</td>
<td>108.00</td>
</tr>
<tr>
<td>Departing passengers’ total time in system (hours)</td>
<td>82 331.24</td>
</tr>
<tr>
<td>Departing passengers’ total walking distance (m)</td>
<td>87 937 718.50</td>
</tr>
<tr>
<td>Number of departing aircraft</td>
<td>1</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Average</td>
<td>2.42</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Maximum</td>
<td>24.99</td>
</tr>
<tr>
<td>Total aircraft circling time (hours)</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft overtime (hours)</td>
<td>54.82</td>
</tr>
<tr>
<td>Total departing passengers’ overtime (hours)</td>
<td>11 130.54</td>
</tr>
<tr>
<td>Aircraft total distance travelled on taxiways (m)</td>
<td>2 761 115.50</td>
</tr>
<tr>
<td>Total number of arriving passengers</td>
<td>135 966.00</td>
</tr>
<tr>
<td>Total number of departing passengers</td>
<td>129 196.00</td>
</tr>
<tr>
<td>Total passenger circling time (hours)</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft time spent taxiing (hours)</td>
<td>141</td>
</tr>
</tbody>
</table>
Table 12.8: Results obtained for Design 4, wind direction 1

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3 (CE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arriving passengers’ total time in system (hours)</td>
<td>46 343.53</td>
<td>45 220.13</td>
<td>44 559.58</td>
</tr>
<tr>
<td>Arriving passengers’ total walking distance (m)</td>
<td>73 291 166.25</td>
<td>70 448 104.75</td>
<td>67 329 879.50</td>
</tr>
<tr>
<td>Number of aircraft in the system: Average</td>
<td>75.07</td>
<td>75.07</td>
<td>75.15</td>
</tr>
<tr>
<td>Number of aircraft in the system: Final value</td>
<td>83.00</td>
<td>83.00</td>
<td>83.00</td>
</tr>
<tr>
<td>Number of aircraft in the system: Maximum</td>
<td>110.00</td>
<td>110.00</td>
<td>109.00</td>
</tr>
<tr>
<td>Departing passengers’ total time in system (hours)</td>
<td>81 643.05</td>
<td>81 591.09</td>
<td>81 837.28</td>
</tr>
<tr>
<td>Departing passengers’ total walking distance (m)</td>
<td>69 845 697.25</td>
<td>67 320 556.50</td>
<td>64 021 959.50</td>
</tr>
<tr>
<td>Number of departing aircraft</td>
<td>1 216.00</td>
<td>1 216.00</td>
<td>1 216.00</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Average</td>
<td>2.41</td>
<td>2.41</td>
<td>2.42</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Maximum</td>
<td>24.97</td>
<td>24.96</td>
<td>24.97</td>
</tr>
<tr>
<td>Total aircraft circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft overtime (hours)</td>
<td>55.72</td>
<td>55.19</td>
<td>63.69</td>
</tr>
<tr>
<td>Total departing passengers’ overtime (hours)</td>
<td>11 059.45</td>
<td>11 035.81</td>
<td>11 178.59</td>
</tr>
<tr>
<td>Aircraft total distance travelled on taxiways (m)</td>
<td>2 699 163.75</td>
<td>2 604 010.75</td>
<td>2 676 547.50</td>
</tr>
<tr>
<td>Total number of arriving passengers</td>
<td>136 003.00</td>
<td>135 966.00</td>
<td>136 155.00</td>
</tr>
<tr>
<td>Total number of departing passengers</td>
<td>129 196.00</td>
<td>129 196.00</td>
<td>129 196.00</td>
</tr>
<tr>
<td>Total passenger circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft time spent taxiing (hours)</td>
<td>116.28</td>
<td>113.59</td>
<td>110.05</td>
</tr>
</tbody>
</table>
Table 12.9: Results obtained for Design 4, wind direction 2

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3 (CE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arriving passengers’ total time in system (hours)</td>
<td>46 603.79</td>
<td>46 010.06</td>
<td>45 207.62</td>
</tr>
<tr>
<td>Arriving passengers’ total walking distance (m)</td>
<td>73 318 546.50</td>
<td>70 577 906.50</td>
<td>66 380 082.75</td>
</tr>
<tr>
<td>Number of aircraft in the system: Average</td>
<td>75.04</td>
<td>74.97</td>
<td>75.16</td>
</tr>
<tr>
<td>Number of aircraft in the system: Final value</td>
<td>83.00</td>
<td>83.00</td>
<td>83.00</td>
</tr>
<tr>
<td>Number of aircraft in the system: Maximum</td>
<td>109.00</td>
<td>109.00</td>
<td>109.00</td>
</tr>
<tr>
<td>Departing passengers’ total time in system (hours)</td>
<td>81 337.56</td>
<td>80 965.17</td>
<td>81 030.26</td>
</tr>
<tr>
<td>Departing passengers’ total walking distance (m)</td>
<td>69 853 519.50</td>
<td>67 438 543.25</td>
<td>63 032 376.00</td>
</tr>
<tr>
<td>Number of departing aircraft</td>
<td>1 216.00</td>
<td>1 216.00</td>
<td>1 216.00</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Average</td>
<td>2.41</td>
<td>2.41</td>
<td>2.42</td>
</tr>
<tr>
<td>Time in system of each aircraft (hours): Maximum</td>
<td>24.95</td>
<td>24.96</td>
<td>24.95</td>
</tr>
<tr>
<td>Total aircraft circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft overtime (hours)</td>
<td>55.43</td>
<td>55.18</td>
<td>64.20</td>
</tr>
<tr>
<td>Total departing passengers’ overtime (hours)</td>
<td>11 042.07</td>
<td>11 033.91</td>
<td>11 173.58</td>
</tr>
<tr>
<td>Aircraft total distance travelled on taxiways (m)</td>
<td>2 698 991.75</td>
<td>2 597 771.75</td>
<td>2 674 130.00</td>
</tr>
<tr>
<td>Total number of arriving passengers</td>
<td>136 003.00</td>
<td>136 003.00</td>
<td>136 155.00</td>
</tr>
<tr>
<td>Total number of departing passengers</td>
<td>129 196.00</td>
<td>129 196.00</td>
<td>129 196.00</td>
</tr>
<tr>
<td>Total passenger circling time (hours)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Total aircraft time spent taxiing (hours)</td>
<td>116.26</td>
<td>113.38</td>
<td>110.15</td>
</tr>
</tbody>
</table>
12.2 The results of the simulation models

12.2.3 Interpretation of the results

Tables 12.10 to 12.13 show these values per passenger or aircraft. The abbreviations in these tables have the following meanings:

- Avg: average
- TIS: time in system
- WD: walking distance
- OT: overtime

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3</th>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg TIS arriving passengers (min)</td>
<td>25.55</td>
<td>24.82</td>
<td>24.13</td>
<td>25.20</td>
<td>24.10</td>
<td>23.34</td>
</tr>
<tr>
<td>Avg WD arriving passengers (m)</td>
<td>877.05</td>
<td>816.39</td>
<td>751.91</td>
<td>876.29</td>
<td>810.30</td>
<td>754.48</td>
</tr>
<tr>
<td>Avg TIS departing passengers (min)</td>
<td>37.93</td>
<td>37.92</td>
<td>37.68</td>
<td>38.29</td>
<td>38.47</td>
<td>38.61</td>
</tr>
<tr>
<td>Avg WD departing passengers (m)</td>
<td>874.43</td>
<td>819.92</td>
<td>750.45</td>
<td>873.42</td>
<td>813.30</td>
<td>753.67</td>
</tr>
<tr>
<td>Avg travelling distance aircraft (m)</td>
<td>2 138.51</td>
<td>2 141.19</td>
<td>2 320.28</td>
<td>2 139.33</td>
<td>2 142.10</td>
<td>2 377.56</td>
</tr>
<tr>
<td>Avg OT departing aircraft (min)</td>
<td>2.95</td>
<td>2.78</td>
<td>3.41</td>
<td>2.95</td>
<td>2.80</td>
<td>3.40</td>
</tr>
</tbody>
</table>
12.2 The results of the simulation models

Table 12.11: Interpretation of results for Design 2

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Wind direction 1</th>
<th>Wind direction 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rule 1</td>
<td>Rule 2</td>
</tr>
<tr>
<td>Avg TIS arriving passengers (min)</td>
<td>20.20</td>
<td>19.30</td>
</tr>
<tr>
<td>Avg WD arriving passengers (m)</td>
<td>512.09</td>
<td>468.79</td>
</tr>
<tr>
<td>Avg TIS departing passengers (min)</td>
<td>37.90</td>
<td>37.76</td>
</tr>
<tr>
<td>Avg WD departing passengers (m)</td>
<td>510.26</td>
<td>470.70</td>
</tr>
<tr>
<td>Avg travelling distance aircraft (m)</td>
<td>2 065.35</td>
<td>1 834.31</td>
</tr>
<tr>
<td>Avg OT departing aircraft (min)</td>
<td>2.51</td>
<td>2.47</td>
</tr>
</tbody>
</table>

Firstly, it is clear that the direction of the wind does not have a substantial influence on the results of the models. Secondly, the different designs are compared based on passenger and aircraft travelling distances and the time spent in the system. It is clear from the results shown that Design 2 is the best when looking at all the performance measures. The average time spent in the system by arriving and departing passengers, the average walking distance of arriving and departing passengers at the airport, the average aircraft travelling distance at the airport as well as the average time an aircraft is delayed (overtime) are the least in Design 2. This design is shown in Figure 12.1. Thirdly, the results from using the different rules are compared:
12.2 The results of the simulation models

<table>
<thead>
<tr>
<th>Design 3</th>
<th>Wind direction 1</th>
<th>Wind direction 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics</td>
<td>Rule 1</td>
<td>Rule 2</td>
</tr>
<tr>
<td>Avg TIS arriving passengers (min)</td>
<td>22.93</td>
<td>22.35</td>
</tr>
<tr>
<td>Avg WD arriving passengers (m)</td>
<td>683.07</td>
<td>624.96</td>
</tr>
<tr>
<td>Avg TIS departing passengers (min)</td>
<td>37.85</td>
<td>37.81</td>
</tr>
<tr>
<td>Avg WD departing passengers (m)</td>
<td>680.64</td>
<td>628.31</td>
</tr>
<tr>
<td>Avg travelling distance aircraft (m)</td>
<td>2 124.38</td>
<td>2 138.26</td>
</tr>
<tr>
<td>Avg OT departing aircraft (min)</td>
<td>2.72</td>
<td>2.61</td>
</tr>
</tbody>
</table>

- It can be seen that by using Rule 2 the walking distance is less than for Rule 1. As stated before, in Rule 1 a small aircraft can only be assigned to a medium or large gate if all the small gates are occupied, and a medium aircraft can only be assigned to a large gate if all the medium gates are occupied. In Rule 2 small aircraft are assigned to medium or large gates that are closer to the terminal building even though there are small gates available further away from the terminal building, and medium aircraft are assigned to large gates closer to the terminal building even though there may be medium gates available.

- The passenger walking distance for arriving and departing passengers in the
### 12.2 The results of the simulation models

#### Table 12.13: Interpretation of results for Design 4

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Wind direction 1</th>
<th>Wind direction 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rule 1</td>
<td>Rule 2</td>
</tr>
<tr>
<td>Avg TIS arriving passengers (min)</td>
<td>20.45</td>
<td>19.96</td>
</tr>
<tr>
<td>Avg WD arriving passengers (m)</td>
<td>538.89</td>
<td>518.13</td>
</tr>
<tr>
<td>Avg TIS departing passengers (min)</td>
<td>37.92</td>
<td>37.89</td>
</tr>
<tr>
<td>Avg WD departing passengers (m)</td>
<td>540.62</td>
<td>521.07</td>
</tr>
<tr>
<td>Avg travelling distance aircraft (m)</td>
<td>2 076.28</td>
<td>2 003.09</td>
</tr>
<tr>
<td>Avg OT departing aircraft (min)</td>
<td>2.75</td>
<td>2.72</td>
</tr>
</tbody>
</table>

models in which the metaheuristic optimisation, i.e. the CE method (Rule 3), is used is smaller than in the models using Rule 1 and Rule 2 of assigning flights to gates. This is the case for each of the four airport apron layout designs to which the generic model was applied. This provides evidence that the CE method, and specifically the generic optimisation model developed in this study, can be used to reduce the passenger walking distances at an airport.

- The time spent in the system by arriving as well as departing passengers is approximately the same for all three rules.
12.2 The results of the simulation models

In general, metaheuristic optimisation, using the CE method, improves the passenger walking distances at an airport. Furthermore, the use of Rule 2 instead of Rule 1 for assigning flights to gates provides better results. The overall best results are produced when Design 2 of the airport apron layouts is used in combination with the generic optimisation model using the CE method (Rule 3) for assigning flights to gates.

12.2.4 The busiest schedule to be used by the airport

The OR Tambo data was scaled using different factors and the resulting schedules were then used to determine at which point the airport became too busy which caused the aircraft to circle the airport while waiting for a gate to become available. When a 1.5 scaling factor was used the total circling time for all the aircraft was still zero, but when a 1.6 scaling factor was used the circling time was bigger than zero. When the 1.5 factor schedule was used the average number of aircraft at the airport was around 75 and the maximum number of aircraft present at the
12.3 Concluding remarks on chapter 12

The data used in the simulation models was based on that of OR Tambo International Airport for a specific time period. However, in order to increase the demand to ensure a busy schedule for the new airport designs, this data was scaled according to a user defined factor. The results show that Design 2 is the best design based on passenger walking distance and time spent in the system as well as aircraft travelling distances on the taxiways. Furthermore, when using metaheuristic optimisation, specifically the CE method, to assign flights to gates, the passenger walking distances are less than when using Rules 1 and 2.
Chapter 13

Conclusion

For the purpose of this study, four different designs in terms of airport apron layout were developed and compared. The layout concept in each design was based on that of Hartsfield Jackson Atlanta International Airport in Georgia, USA. However, the orientation of the terminal building with respect to the concourses was altered in each design. The study was done in collaboration with Virtual Consulting Engineers in Pretoria to find a good future layout for Lanseria International Airport.

The aim of the study was to minimise passenger walking distances and waiting times at an airport. This was done by improving the apron layout and the aircraft gate assignment process. For this purpose, it was determined which of the four airport layouts would be the best to use at Lanseria International Airport. Furthermore it was decided whether the use of the CE method to assign flights to gates would reduce the passenger walking distance.

The literature review that was done for the purpose of the study included a background of the operations and components of an airport, an overview of simulation and the cross-entropy method as problem solving techniques as well as previous work done in this regard in the form of two case studies. Simulation was identified to be a suitable technique for solving the problem, since the dynamic nature and the complex factors of the study could be thoroughly addressed. These factors included delays resulting from boarding processes not being completed before the departure time of a flight had been reached, aircraft that had to wait at the end of a taxiway for other aircraft to pass in order to avoid collisions, and
the state of the gates that changed from available to occupied over time. An important factor that had to be considered was to which gate each flight was assigned. This had a large influence on the average passenger walking distance. A metaheuristic optimisation process, using the CE method, was designed to calculate a flight-to-gate assignment schedule in which the passenger walking distances would be reduced.

The problem solving phase of the study was twofold. In the first part, a simulation study was done to compare the different airport designs. A concept model of the airport operations considered in the study was developed. The simulation models were then built and run using an arrival and departure schedule of OR Tambo International Airport. This schedule was adjusted to suit the new airport by using a scaling program. Different validation and verification methods were used throughout the development of the models. The performance measures used in the comparison process included 1) the average walking distance for arriving and departing passengers, 2) the average time spent at the airport by arriving and departing passengers, 3) the average time by which a flight was delayed, 4) the average distance travelled by an aircraft at the airport and 5) the average number of aircraft present at the airport.

In the second part of the problem solving phase, a generic optimisation process was designed to calculate a flight-to-gate assignment schedule in order to reduce the average passenger walking distance. The cross-entropy method was used for this purpose. The optimisation process was implemented within the simulation software and was performed in real time, thus a new schedule was calculated each time a delay occurred. Previous use of the cross-entropy method, to the extent that could be ascertained, does not include flight-to-gate assignments at an airport or real time application of the method. However, evidence was provided in this study that the method can be used for both of these situations.

In the analysis phase of the study, different scenarios were compared. Each of the four airport designs were run with the wind direction being from either side of the runways. Each of these were then run, firstly by using Rule 1 in assigning flights to gates, secondly by using Rule 2 and thirdly by using Rule 3. In Rule 1, an aircraft was assigned to the available, suitable gate closest to the terminal building, while only assigning small aircraft to medium and large gates when no
small gates were available and assigning medium aircraft to large gates only when no medium gates were available. In Rule 2, an aircraft was also assigned to the available, suitable gate closest to the terminal building, but in this rule small aircraft could be assigned to medium and large gates even if there were small gates available and medium aircraft could be assigned to large gates even if there were medium gates available. In Rule 3 the generic optimisation process, using the CE method, was used for finding the optimal flight-to-gate assignment.

The results revealed the following:

- The direction of the wind does not have a substantial influence on the performance measures used in the models.
- The second design in terms of airport layout is the best when considering the aforementioned performance measures.
- When using Rule 2, the average walking distance per passenger is less than when using Rule 1.
- When using the optimisation process, i.e. the CE method (Rule 3) to assign flights to gates the average passenger walking distance is less than when using rules one or two.
- The cross-entropy method can be dynamically used for simulation problems.
- In order to ensure that the arrival and departure schedule used at the airport is not too busy, the average number of aircraft at the airport must be kept below 75 with the maximum number of aircraft below 110 when using any of the developed rules for assigning flights to gates.

The objectives of the study, as stipulated in chapter 1, were met. Four different apron layouts were designed and compared using computer simulation. Three rules/algorithms were developed for assigning flights to gates. A generic optimisation process, using metaheuristic optimisation and specifically the CE method, was developed as one of the rules. This process can be applied to any airport model to reduce the passenger walking distances. It was concluded that the second airport design will be the best to use at Lanseria International Airport.
and that the use of metaheuristic optimisation, and specifically the cross-entropy method, for assigning flights to gates does reduce the passenger walking distances at an airport. These solutions were found to be viable and can be used in the expansion of Lanseria International Airport. Furthermore, it was proved that the cross-entropy method can be applied in real time to simulation problems.

It is recommended that, in terms of operational efficiency, Design 2 should be used in the expansion of Lanseria International Airport and that the flight-to-gate assignment processes at the airport should be performed using the developed optimisation process which uses metaheuristic optimisation and specifically the CE method. However, future work could involve a cost analysis of the development of each of the four airport apron designs in order to determine whether Design 2 would be feasible in this regard.
References


ATLANTA AIRPORT (2007). Year-to-date passenger data. Tech. rep., Hartsfield Jackson Atlanta International Airport. 33


REFERENCES


BOEING (2010). 767 airplane characteristics for airport planning. Tech. rep., Boeing. 72


REFERENCES


REFERENCES


OFFERMAN, H. (2001). Simulation to support the airport stakeholder decision-making process. *Air & Space Europe, 3,* 60 – 67. 11, 12


REFERENCES


REFERENCES


