An integrated approach for efficient and resilient airport management

MSc Thesis

J. Spans

Delft University of Technology

TU Delft

An integrated approach for efficient and resilient airport management

MSc Thesis

by

J. Spans

in partial fulfilment of the requirements for the degree of

Master of Science
in Aerospace Engineering
at the Delft University of Technology.

Student number:	4170113	
Date:	July 5, 2018	
Version:	Final	
Thesis committee:	Dr. A. Sharpanskykh,	Delft University of Technology, supervisor
	Dr. Ir. H. G. Visser,	Delft University of Technology, supervisor
	Dr. M. A. Mitici,	Delft University of Technology, supervisor

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Abstract

One of the main research areas in air transportation is the area of flight-to-gate assignment problems. In flight-to-gate assignment problems, flights are assigned to gates such that the utilities of all stakeholders are maximised. Among the stakeholders are the passengers, whose preferences are up till now often represented in terms of passenger walking distances (distance between entrance and gate). However, the passenger experience is not only affected by the distance they need to walk but also by congestions it encounters at passenger processing points.

One of the most promising modelling techniques, that is capable of modelling airport passenger flows and terminal processes, is the agent-based modelling and simulation paradigm. Agent-based modelling and simulation takes a bottom-up approach. This approach aims to identify emerging patterns at the overall group level, by creating agents with specific capabilities at local level.

Currently, the airside and landside of an airport are independently managed. This also reflects the scientific gap that is existent in current literature. Therefore, the current study aims to open a new research area by integrating the two paradigms described above. Such that the integrated approach enjoys the strengths of both paradigms. The main research objective is as follows:

To develop a methodology to integrate an agent-based model for airport terminal processes with a flight-to-gate assignment optimisation, such that these airside and landside processes can be managed simultaneously.

The current research will mainly focus on the security checkpoint queue, since it was found that the security checkpoint is the main landside source of airside delay. Two integration methodologies have been created in this research. The first integrates the agent-based model directly into the flight-to-gate assignment optimisation strategy (direct integration). The second methodology aims to approximate the agent-based model response by means of meta-models (indirect integration), before integrating the meta-models with the flight-to-gate assignment optimisation.

The objective of the integrated optimisation is to make sure that the passenger experience at the security checkpoint is optimised. The passenger experience is assumed to be optimal when the spread of experienced queue times among the different security checkpoints of the airport is minimised. This can be achieved by assigning the flights-to-gates such that passengers departing with a specific flight take a roughly defined path to the assigned gate. The problem is optimised using a differential evolution algorithm.

The direct integration methodology, based on the simulation optimisation framework, integrates the agent-based model directly into the optimisation routine. An advantage of the proposed methodology is the fact that there is little (to no) loss of detail of the agent-based model and the found optimal assignment is assumed to be the actual optimal assignment. However, the major disadvantage of this method is the fact that the optimisation of a small case study (considering 8 flights and two gates) could take in the order of days to complete. This is partly due to the fact that each scenario (flight-to-gate assignment) needs to be simulated multiple times using the agent-based model, since each simulation run could result in a different realisation. Furthermore, the optimisation algorithm used, explores both feasible and infeasible solutions. Especially these infeasible solutions can be challenging in terms of simulation time.

The indirect integration methodology, using meta-models created of the governing agent-based model, is able to optimise in the order of minutes. However, the validity of the found optimal assignment is dependent on the quality of the constructed meta-models. Two meta-model types have been created: Regression meta-models and Gaussian radial basis function meta-models. Both meta-model types

have difficulty approximating the highly non-linear behaviour apparent in the agent-based model data (high approximation errors were observed). Furthermore, by means of a validation case study it was observed that the indirect integration methodology was not yet able to locate the same optimal assignment as the direct integration methodology.

Current research has opened up a whole new research area by developing two novel methodologies to integrate an agent-based model and simulation with a flight-to-gate assignment optimisation. The challenge during the integration was to match the level of the agent-based model (micro/passenger level) with the level of the flight-to-gate assignment optimisation (macro/airport level), without losing too much detail. The scientific contribution is substantial, since up till now the integration of an agent-based model with a flight-to-gate assignment optimisation has not yet been investigated nor performed. In addition, this study describes in detail all the steps taken and explored necessary to make the integration possible, which can be confidently used by future researchers.

This research can be further improved by focussing on improving the meta-models' performances. Furthermore, future studies could investigate the real-time reassignment problem, by creating a coordinating agent that reassigns flights-to-gates when congestions occur in the terminal or flights are delayed. Another suggestion would be to further calibrate the agent-based model and simulation. In the future, the developed methodologies could be used in real practice. Airport managers could use the methodologies to forecast passenger flows and pro-actively avoid congestions at the airport.

Preface

Before you lies my thesis “An integrated approach for efficient and resilient airport management”, the novel research that studies the integration of an agent-based model and simulation with a flight-to-gate assignment optimisation. It has been written to fulfil the requirements for the degree of Master of Science in Aerospace Engineering at the Delft University of Technology. I was full-time engaged in researching and writing this thesis for around 9 months.

I would really like to thank my supervisors Dr. O.A. Sharpanskykh, Dr.Ir. H.G. Visser and Dr. M.A. Mitici for their excellent guidance and support during this process. Furthermore, I would like to thank Stef Janssen for the pleasant cooperation that we had during my thesis.

Last but not least, I would like to thank my family and friends for their continuous support. First of all, my girlfriend Amber who took the time and courage to review my thesis and keep me motivated during the more difficult periods. Finally, I would like to thank my parents and brother for supporting me during this final year, but also throughout my entire study period.

J. Spans
Delft, July 2018

Contents

List of figures

List of tables

List of acronyms

Acronym	Description
AAS	Amsterdam airport Schiphol
AATOM	Agent-based airport terminal operations model
ABM	Agent-based model
ABMS	Agent-based modelling and simulation
ACI	Airports council international
AR	Arrival rate
CA	Colonisation algorithms
DE	Differential evolution algorithm
FGAP	Flight-to-gate assignment problem
GA	Genetic algorithm
GRBF	Gaussian radial basis function
GSA	Global sensitivity analysis
IAT	Inter-arrival time
IATA	International Air Transport Association
IED	Improvised explosive device
IQR	Interquartile range
KG	Kriging method
KPI	Key performance indicators
LoS	Level of Service
MC	Monte Carlo
MSE	Mean Squared Error
ODE	Ordinary differential equation
OR	Operations research
PDE	Partial differential equation
RC	Representational capability
RMSE	Root-mean square error
RSM	Response surface methodology
RTHA	Rotterdam The Hague Airport
SA	Simulated annealing
SC	Security checkpoint
SCQT	Security checkpoint queue time
SE	Standard error
SG	Shin Goel algorithm
STD	Standard terminal departure
TS	Tabu search

1 Introduction

Since the Airline Deregulation Act of 1978, air transport has become a highly competitive market. This has initiated a trend towards lower ticket prices, making air transport available to the general public. Demand for air travel is growing ever faster, which means more flights per hour at airports, bigger aircraft, and a general higher passenger density. Managing these streams of traffic is one of the challenges airport managers face on a daily basis. A main field of research to relieve the airport managers is the flight-to-gate assignment optimisation. Many factors are taken into account when considering flight-to-gate assignments, such as the minimisation of passenger walking distances.

However, in reality these flight-to-gate assignments are not deterministic. Delays can be caused by various reasons such as weather delays, aircraft congestion and technical problems. In addition, there can be terminal delays that affect the passengers at the airport. May 2017, Amsterdam Airport Schiphol (AAS) became so congested, that passengers were missing their flights due to congestions at check-in points, passport control and security checkpoints. The latter was one of the biggest bottlenecks. Furthermore, there is a growth in passengers bringing checked luggage, which makes check-in queues longer. In general, passenger numbers will continue to grow. In April 2018, 6 million passengers flew to or via AAS, which is 3% more than the previous year (Royal Schiphol Group Mediarelaties (2018)). It is expected that passenger growth will continue in the coming years. Improved infrastructure is needed to accommodate the growth. However, investment in expansion of airports is costly and does not always dissolve bottlenecks as was the case at AAS. Therefore, new smart approaches have to be developed that could be used to cope with the growth in passenger traffic at airports.

One of the research fields that is interested in modelling passenger flows at airports is the agent-based modelling and simulation paradigm. Properly calibrated and validated agent-based models and simulations can be used to assess infrastructure changes or schedule scenarios at real airports. To understand and use passenger flow knowledge at airports, a novel research is proposed to investigate the integration of flight-to-gate assignments model with a passenger flow model. Such an integrated model could be used to manage landside and airside simultaneously and possibly prevent cases like the one from AAS. The passenger flows at airports will be represented by an agent-based model and simulation (ABMS), which is a powerful method to study complex systems and has proven to be very useful in traffic and transportation systems (Chen (2010)). However, in general these agent-based models are computationally heavy. Hence, these models are less preferred for quick scenario assessment.

Therefore, both direct and indirect integration methods will be developed. The direct integration method will integrate an agent-based model directly into the flight-to-gate assignment optimisation routine. In the indirect integration method, agent-based model abstractions will be used to integrate with a flight-to-gate assignment model. Once these integrated models of both landside and airside processes are built, scenarios will be analysed to show the added value of such an integrated approach for efficient and resilient airport management. Results of this research might also be used in, or useful for, current airport operations.

This report contains every step of the research performed and explored. In chapter 2 the most important literature for the current research is reviewed. In addition, the existing scientific gap is identified. Chapter 3 elaborates on the research framework, by defining the problem statement, research objective and questions. The chapter is concluded with the methodology that will be used to achieve the research objective. In chapter 4 the adjustments that have been made to the agent-based airport terminal operations model (AATOM) will be discussed. Then, in chapter 5 the direct method of integrating the agent-based model with a gate and resource assignment optimisation will be treated. The problem will be mathematically formulated and the integration method developed. Chapter 6 introduces the second indirect method of integration by meta-modelling. Before the actual integration can take place, the abstractions of the agent-based model need to be created. The methods of integration are tested in two case studies in chapter 7. In chapter 8 the results are discussed and implications are given. Finally, the study is concluded in chapter 9, where the research questions are answered and recommendations for future research are given.

2
Literature review

In this chapter a literature review is presented. The main goals are to summarise what research has been done and present the state-of-the-art in relevant scientific fields. In the literature review report "An integrated approach for efficient and resilient airport management: Literature review" Spans (2017), previously performed research was identified that could be used as input for this thesis. The current section will give an overview of the most important and influencing findings of that literature review. Since this thesis aims to open a whole new research direction, the current literature review will zoom in on the separate parts involved that will be needed for the integration in the body of this thesis.

This chapter has been split into three main sections: 1. Flight-to-gate assignment problems (section 2.1), 2. Airport passenger flow modelling (section 2.2), 3. Simulation optimisation. Finally, in section 2.4 a main conclusion is drawn from the literature reviewed.

2.1. The flight-to-gate assignment problem

In this section the research into flight-to-gate assignment problems (FGAP) is reviewed. This section starts off by looking at the problem description of FGAP, in section 2.1.1. In the same section, the parties involved in a FGAP will be briefly addressed and their interests given. In section 2.1.2, the models developed to represent the FGAP are classified according to their characteristics. The second part of this section, starts in section 2.1.3 and is concerned with the solution methods used in FGAPs.

2.1.1. Problem description

Air transport growth has driven research into techniques for managing and allocating airport and airline resources in a dynamic operational environment. Increasing passenger demand for comfort, in combination with fierce competition between airlines, have led to the need for new models and methods (Dorndorf et al. (2005)). One of the most complex and important problems that operations managers daily face is the flight-to-gate assignment problem (Bouras et al. (2014)).

The problem that these managers try to solve is which aircraft to assign to which gate while maximising for passenger convenience, airline preferences and/or airport efficiency. Dorndorf et al. (2005) argues that the problem these managers face today are more complicated than most other traditional scheduling problems. This can be attributed to two factors. Firstly, due to the high number of factors that have to be taken into account such as incoming/outgoing flights, the presence of transferring passengers, terminal equipment, crews and baggage handling. In addition, the use of available resources is highly interdependent. Therefore, a resource management system for airports of any size is a complex task.

Assignment considerations and objectives

A solution to the FGAP, is such that the preferences of all stakeholders are fulfilled up to a satisfactory level. Clearly, each stakeholder wants its own utility to be maximised. However, to ensure a durable system the gain needs to be spread among parties involved. Besides preferences of the stakeholders, there are also restrictions imposed by regulations and facilities which are therefore of general interest.

These imposed restrictions define feasible solutions. Hence, if one of the conditions has not been met, then the flight-allocation is not feasible. This could for example be that the gate or parking position should technically fit the aircraft. Furthermore, certain origin destinations of a flight require a custom's facility. So if this is not in place for a certain gate then it is not a feasible allocation.

Papers on the FGAP describe similar desires of the different parties involved. Research by van Rhee (1992) describes the preferences of passengers, ground handlers, airlines, and the airport. Some example assignment considerations as given by van Rhee are given below.

Passengers' preferences

- Gates with jet bridges are preferred over off-gate parking places.
- Waiting rooms should be suitable for the number of departing passengers in a flight.
- Arriving/departing passengers prefer short walking distances towards the exit/entrance.
- Connecting passengers prefer short walking distances towards their connecting flight.

Ground handlers' preferences

- Acceptable transport distances for luggage and handling equipment.
- Parking spaces with enough clearance.
- Successive flights of one airline parked at the same position or in the vicinity.
- Grouping of aircraft, such that multiple aircraft can be handled at the same time.

Airlines' preferences

- Minimal handling costs.
- Of utmost importance is the fulfilment of passenger' interests.
- Requests for special parking positions should be honoured if possible.

Airport preferences

- Utilities of all parties involved should be proportionally maximised.
- Minimal operating costs.
- A balanced passenger and aircraft spread among the airport.

A vast amount of papers have tried to solve the FGAP, whilst taking the above preferences into account. However, creating an optimised gate assignment plan involves a certain objective. A clear overview of proposed objectives in the FGAP is given by Aktel et al. (2017). The majority of the papers on gate assignments, propose an objective function that minimises walking distance of the passengers. One of the first papers written on FGAPs is by Babić et al. (1984). In this paper aircraft are allocated to an aircraft stand while minimising passenger walking distances. For arriving passengers, the walking distance is the measured distance between the gate and (central) baggage claim area. The walking distance for departing passengers is measured between check-in counters and gates.

Similar to the paper by Babić et al. is the work by Mangoubi and Mathaisel (1985) which tries to minimise passenger walking distances within the airport terminal using data from Toronto National Airport. In addition to arriving and departing passengers, walking distances of transferring passengers are also taken into account. The walking distance for transferring passengers is dependent on the location of both the arriving gate and departing gate. However, the connecting gate of transferring passenger is not known at that time. Therefore Mangoubi and Mathaisel make use of an uniform distribution assumption to determine all inter-gate walking distances (the average walking distance to all other gates). Mentioned in the paper is that this assumption has some major drawbacks, as in reality the distribution is not uniform. Finally, the assignment problem is solved using a linear programming relaxation of an integer program formulation and a heuristic. The latter proved to significantly reduce the computational time if used as starting point. The model presented by Mangoubi and Mathaisel is a simple model which could perfectly serve as a basis to build from.

Other papers that try to solve the FGAP using passenger related objectives include the work by Chang (1996), Ding et al. (2004), Haghani and Chen (1997), Xu and Bailey (2001), Yan and Huo (2001).

2.1.2. Flight-to-gate assignment problem classification

In this sub-section, some structure is added to the vast amount of literature on the FGAP. This is done by classifying the models used for the FGAP into deterministic and non-deterministic models.

According to Cheng et al. (2012), the models used for the flight-to-gate assignment can be classified into static models and stochastic & robust models. The main difference between the two types of models is that the static model deals with a deterministic formulation. Whereas, the stochastic and robust models take disruptions into account and do not rely on an deterministic world (ideal world). The main difference between stochastic and robust models, is the fact that stochastic models start by assuming that the uncertainty has a probabilistic description whereas in robust optimisation the uncertainty converted into deterministic and set-based buffers. Therefore, the decision is made to classify the models either as deterministic or non-deterministic.

Deterministic models

The first class of models are the deterministic models. Especially in the early years of research into FGAPs, deterministic models were the way to go. All the papers by Babić et al. (1984), Ding et al. (2004), Haghani and Chen (1997), Mangoubi and Mathaisel (1985), Xu and Bailey (2001), Yan and Huo (2001) propose deterministic gate assignment models. These models assume a fixed flight schedule and a fixed number of gates. The main assumption of these models is that the expectations of the systems are equal to the realisations, there is no uncertainty.

Recall that Mangoubi and Mathaisel used a model to minimise the passenger walking distance, taking into account arriving, departing and transferring passengers. The model proposed can be classified as a deterministic model.

The parameters used are the following:

- N is the set of flights considered,
- M is the set of gates considered,
- n is the total number of flights,
- m is the total number of gates,
- p_i^a, p_i^d, p_i^t denote the estimated number of arriving, departing and transferring passengers using flight i, respectively,
- d_j^a, d_j^d and d_j^t denote the distances which the three types of passengers need to bridge for respectively arriving, departing and transferring passengers from gate j,

The binary decision variable x_{ij} is assigned for each possible flight-to-gate assignment, where:

$$x_{ij} = \begin{cases} 1 & \text{if flight i is assigned to gate j,} \\ 0 & \text{otherwise.} \end{cases}$$

In eqs. (2.1) to (2.3) the deterministic model of Mangoubi and Mathaisel (1985) is shown. The total walking distances for all passengers is denoted by Z.

$$\min Z = \sum_{i=1}^{n} \sum_{j=1}^{m} (p_i^a d_j^a + p_i^d d_j^d + p_i^t d_j^t) x_{ij}, \tag{2.1}$$

s.t.

$$\sum_{j=1}^{M} x_{ij} = 1 \quad \forall i = 1, ..., N \tag{2.2}$$

$$\sum_{h \in L(i)} x_{hj} + x_{ij} \leq 1 \quad \forall i = 1, 2, ..., N \quad \forall j = 1, 2, ..., M \tag{2.3}$$

The constraint in eq. (2.2) ensures that every flight is assigned to one and only one gate. Furthermore, eq. (2.3) makes sure that two aircraft may not be assigned to the same gate concurrently. $L(i)$ is the set of all flights h which landed before flight i and are still on the ground at the time flight i arrives. $L(i)$ is termed a “conflict set”. As one can see, there is no uncertainty included in the model itself. Hence, the deterministic type of model assumes an ideal world.

However, airport operators produce the gate assignment plan one-day in advance. During the actual day deviations from original plan might occur. Assuming that the initial one-day ahead plan was made using a deterministic model, then this calls te need for real-time re-assignments. As in the deterministic model these unforeseen events where not taken into account. Therefore, the second class of models are the non-deterministic models.

Non-deterministic models
The first subcategory of the non-deterministic models are the stochastic models for the FGAP. This optimisation problem is based on probabilistic information. Gu and Chung (1999) try to model the effect of stochastic flight delays, focussing on the reassignment problem. This paper introduces a genetic algorithm to solve the FGAP and finds the best assignment with regards to the extra delay time.

In the work by Yan and Tang (2007) not only stochastic time delays are taken into account, but also incorporate the probabilities of the delays into the model. Yan and Tang create a stochastic flight delay gate assignment model. This model is a gate assignment model which takes flight delays into account using data from Taiwan international airport. The model was formulated as an integer multiple commodity network flow program and made use of expected semi-deviation risk measure.

The second subcategory concerns the robust gate assignment models. The robustness of the model relates to the ability of an assignment plan to remain sustainable under minor disturbances in the scheduled flight departure and arrival times. Already in the paper by Mangoubi and Mathaisel (1985) a fixed buffer was suggested between two successive flight, which is an attempt to create a robust plan. The fixed time buffer allowed for deviations in flight schedules.

Flight delays, severe weather, or equipment failures could potentially disrupt planned schedules and therefore Bolat (2000) produces a mixed-binary mathematical model with a quadratic function for minimizing the variance of idle times at the gates. Idle time is the time between two successive flights in which the gate is not being used (empty). When an aircraft departs late from its gate, it does not directly disrupt the pre-made assignment schedule. This would only mean that the planned idle time after departure is partially used. Hence, each flight should be assigned to a gate such that the idle time after its departure is maximised. However, the assumed ground time of flights and the available time of gates are constant. Therefore the goal of maximising the idle time after departure can only be achieved by uniformly distributing the idle times over the gates. As a result the probability of gate conflicts is minimised.

In the paper by Diepen et al. (2012), a gate assignment plan is made specifically for Amsterdam Airport Schiphol (AAS). The model that is based on *gate plans*. Each *gate plan* consists of a subset of the flights grouped because they can be assigned to a single gate of a certain type. Gates are then grouped according to their type. Finally, the best subset of gate plans is selected such that each flight belongs to one selected gate plan, and such that the number of selected gate plans for a certain type of gate is equal to the number of gates of this type. In order to maximise robustness, the idle time between consecutive flights at a gate is maximised.

Reducing the prevalence and impact of gate blockage is another way of creating a robust assignment. Gate blockage is the occurrence of an aircraft which is on-time and assigned at a certain gate but has to wait because the preceding aircraft is still occupying the gate. Castaing et al. (2016) formulate an optimisation problem that assigns flights to gates so as to minimise the expected impact of gate blockage using historical data to predict delay distributions. Li (2009) uses a probability distribution function on gate conflicts between two aircraft to minimise the number of gate conflicts.

Finally, the paper by van Schaijk and Visser (2017) proposes a novel robust solution to the FGAP. Deterministic gate constraints are replaced by stochastic gate constraints, which include stochastic flight delays. In the paper a regression is made in order to predict flight presence probabilities whilst taking into account specific explanatory variables. This is a well established model, however the model requires specific data which might require restricted access.

The robust models are an improvement compared to the deterministic models, as they account for disturbances to the existing flight plans. The world represented in robust gate assignment models is closer to reality. However, most of the resulting plans exhibit one-sided robustness, due to the fact that only flight delays are taken into account. However, an airport managers could also deviate from the gate assignment plan when disturbances at airport areas are observed. The work by Şeker and Noyan (2012) shows an initial attempt to model terminal originating delays. The paper introduces an estimation function on the gate conflict. A similar approach is used in the paper by Lim and Wang (2005).

Another way to classify the models by single or multiple time slot models. The review of the related literature is not shown here, but can be found in Spans (2017).

2.1.3. Solution methods for the flight-to-gate assignment models

In this sub-section the several solution methods used to solve the FGAP will be discussed. The methods can (roughly) be split up into three groups: Expert methods, Simulation methods, and Mathematical programming techniques. The methods will briefly be explained and important papers will be referred to.

Expert methods

One way to solve a FGAP is with the use of expert based systems. Expert systems are a branch of applied artificial intelligence, and were developed by artificial intelligence researchers in the mid-1960s. The idea behind expert systems is that expertise, task-specific knowledge of humans, is recreated in a computer as a model/software. End-users of the model could then call upon the computer for specific advice. The expert system can make inferences, before arriving at a conclusion. As like a human advisor, the system gives an advice and explains, if necessary, the logic behind the advice (Aronson et al. (2005)). The logic behind the given advice is dependent on the rules implemented in the system. Depending on the system, system operators are or are not in the position to change or add rules in order to improve the system.

For the FGAP, Gosling (1990) argues that traditional operations research techniques have difficulty dealing with uncertain information and multiple performance criteria. Furthermore, the operations research techniques do not fit the needs of real-time operations support. A lot of research has focussed on rule-based expert systems and papers like Brazile and Swigger (1988), Su and Srihari (1993) use expert models to solve the FGAPs.

Simulation methods

Another way to solve the FGAP is by making use of so called simulation methods. Obviously, simulation methods use a simulation approach to solve the problem. This method is most often used in the situation where the deterministic gate assignment does not hold any more and the operations managers need to interfere. This could be the case when flights are delayed or when there are bottlenecks of passenger flows in the terminals. It can be seen as a planning and operational tool for simulating the assignment of gates to aircraft. It can be used to evaluate the effectiveness of operational options to improve the gate utilisation.

In the study conducted by Yan et al. (2002), a simulation framework is used that is able to analyse the effects of stochastic flight delays on deterministic assignments. In addition, the simulation can also be used to evaluate flexible buffer times and real-time gate assignment rules based on greedy heuristics. Another paper which solves the FGAP using a simulation method is Hamzawi (1986). The model used in the microcomputer is expert-knowledge based.

Mathematical programming techniques

The mathematical programming techniques can also be seen as the optimisation methods and can be further subdivided into *1.) Exact methods* and *2.) (Meta-) Heuristic methods*. Exact methods could make use of branch and bound in combination with the simplex method to arrive at the exact optimal solution. Heuristic methods can also be used if the global optimum is not of major concern. These methods mostly search for a satisfying solution.

The work of Babić et al. (1984), Bolat (2000) use the branch and bound techniques to minimise the number of passengers who have to walk the maximum distances, and minimise the variance of the gate idle times, respectively. Yan and Huo (2001) use a multi-objective integer programming, simplex method with column generation and branch and bound techniques for the minimisation of passengers' walking distances and waiting times. The column generation approach is conventionally used for solving linear problems, whose optimal solutions may not be integers. Therefore, column generation needed to be used together with the branch and bound technique to deal with larger scale problems.

A drawback of the exact algorithms is the fact that the computational time rapidly increases when the problem size increases. To overcome the problem of computational time, heuristic methods can be used. Mangoubi and Mathaisel (1985) used a combination of two mathematical approaches, a heuristic and a linear programming relaxation, to find the solution that minimises transferring passengers' walking distances. No integer programming algorithm was needed since the linear programming relaxation already led to a 0, 1 optimal solution. It was found that the computational time of the linear programming relaxation was substantial, when compared to the heuristic approach. However, the computational time could substantially be decreased by using the heuristic solution as the starting point for the linear programming relaxation.

When exact optimization fails to generate a solution in an acceptable amount of time, or fails to have a solution at all, or the objective function is non-linear , heuristics can be used. Heuristic methods can be divided into meta-heuristics and problem specific heuristics.

Problem specific heuristics can be interpreted as a set of rules (defined using expert-knowledge) to limit the solution space in order to speed up the solution process. Specific heuristics are developed in the work of Mangoubi and Mathaisel (1985) and Haghani and Chen (1997).

Meta-heuristics is becoming the more popular way to solve complex problems. Commonly used meta heuristics include tabu search (TS), simulated annealing (SA), genetic algorithm (GA), and various types of colonisation algorithms (CA). In addition, a promising algorithm is the differential evolution (DE) algorithm that is supposed to converge faster and with more certainty than many other acclaimed global optimisation methods.

A tabu search algorithm is a very effective tool in many optimisation problems. A TS approach searches for the optimal solution with help of an adaptive memory procedure (Vemuganti (1999)). Each iteration the algorithm evaluates neighbourhood moves. The algorithm selects one, and then moves from the current solution to a new solution. Restrictions are imposed to classify certain moves tabu and thus restrict their selection. A non-tabu move with the highest evaluation is conventionally selected, although aspiration criteria permit sufficiently attractive moves to be selected in spite of their tabu status. After a pre-specified amount of iterations, the algorithm stops. Haghani and Chen (1997) used a tabu search algorithm to solve a single timeslot mixed integer quadratic gate assignment problem. As explained, the algorithm exploits the special properties of different types of neighbourhood moves and creates effective candidate list strategies. Finally, all feasible solution sets were examined, and the best solution for smallest total walking distance was chosen. In the same paper, the use of the heuristic method was compared with the solution obtained with CPLEX (commercial software). It was shown that TS obtained the same optimal solution in less computational time. Ding et al. (2004) adjusted the model used by Haghani and Chen (1997) to allow it to handle over-constrained flight schedules with the objective to minimise the amount of unassigned flights and passenger connection times.

Simulated annealing is a generic probabilistic heuristic approach. SA locates a "sufficient" approximation of the global optimum of a given objective function in a large sample space. SA considers just like the TS algorithm neighbouring solutions and compares them to the current solution. The SA probabilistically chooses either to accept a new solution or to keep the current. The probabilities are tuned in such a way that the problem ultimately tends to the solution with a better objective value. SA is used in the multicriteria airport gate assignment of Drexl and Nikulin (2008).

A genetic algorithm as used by Bolat (2001), is originally developed by Holland (1975). GA provides a group of generations (solutions), which are evolved towards better solutions using biological-inspired operations such as mutation, cross-over and natural selection. The work by Gu and Chung (1999) shows an application of the genetic algorithm to solve the aircraft gate reassignment problem. It was found that the genetic algorithm was well capable of finding the global optimum.

Colony based meta-heuristics is a relatively new means to solve a optimisation problem. An example of a CA is in the work by Pintea et al. (2008). In this study the FGAP is solved with a hybrid ant system. The algorithm is based on the way ants search for food and find their way back to the hive. They do this with the use of pheromone trails. Other agents (ants) are attracted by the pheromone, so they choose the path with a greater accumulation of pheromones at junctions. Random factors are inserted to avoid inflexible solutions in a dynamic environment. It was shown in this paper that the CA lead to the optimal solution.

The most promising technique currently being used is the differential evolution algorithm. This algorithm is acclaimed to converge faster and with more certainty than other global optimisation methods over continuous spaces (Storn and Price (1997)). This paper shows the performance of the algorithm by means of an extensive test bed. Several other algorithms were compared to the DE such as: Adaptive Simulated Annealing, the Annealed Nelder and Mead approach, the Breeder Genetic Algorithm, The EASY Evolution Strategy and the method of Stochastic Differential Equations. All of the above algorithms were outperformed by the DE in terms of required number of function evaluations necessary to arrive at the global optimum of a test function. This was a big breakthrough at that time, since the DE uses a fairly simple and straight forwards optimisation strategy.

First, the algorithm initialises a random population with (a to be specified) number of individuals such that most of the parameter space is covered (target vector). Secondly, these initial population is evaluated by the fitness function. Then the DE mutates the target population to a new population by adding the weighted difference between two population individuals to a third individual. Then, crossover takes place which increases the diversity of the perturbed parameters. The crossover process output will be a trial solution that will challenge the target vector.

What should be noted is that most of the meta-heuristics perform way on a continuous parameter space. Since the idea is to define and create a FGAP such as defined by Mangoubi and Mathaisel, it might be problematic to use these meta-heuristics. However, Pan et al. (2007) proposes a discrete differential evolution (DDE) algorithm which is capable of solving a scheduling problem. Pan et al. found that the DDE was still competitive to other algorithms such as the iterated greedy algorithm for discrete parameter space problems.

2.2. Airport passenger flow modelling

In 2017 Amsterdam Airport Schiphol (AAS) saw over 68 million passengers strolling around the airport. Entering, leaving or connecting from one gate to the other. The busiest airport is Atlanta International Airport, which saw over 103 million passengers in 2017. Air transport is expected to double, to over 14 billion by 2029 based on ACI's forecast annualized growth rate of 4.9% (Airports Council International (2016)). This means that passenger guidance at airports is becoming ever more important. From the moment a passenger arrives at the airport their progress, speed and dwell-time can all be used to smooth the flow of people through the terminal. The main airport issues and challenges are concerned with capacity, congestion and delay. The capacity of an airport is the ability to accommodate a given level of traffic. Where traffic could be aircraft, people, luggage, freight or vehicles. When the capacity reaches its limits or is exceeded, congestion occurs. These congestions, an accumulation of e.g. passengers, could cause disorders or even delays.

This section starts off by looking at the distinct airport passenger processes section 2.2.1. These processes can be modelled using the agent-based modelling and simulation (ABMS) paradigm. The general working of agent-based modelling and simulation is explained in section 2.2.2. Furthermore, existing literature on the application of ABMS in modelling passenger flows is discussed. Section 2.2.5 discusses the difficulties encountered when validating an agent-based model and simulation.

2.2.1. Airport passenger processes

At airports, passengers arrive and depart at both airside and landside. This is schematically shown in figure 2.1. The top flow are the passengers that arrive at the airside (arriving/transferring passengers) and on the bottom are the passengers that arrive at the landside (departing passengers). The figure shows the main areas or processing points that the passenger will encounter. Not all passengers pass the same areas, this depends on the type of passenger.

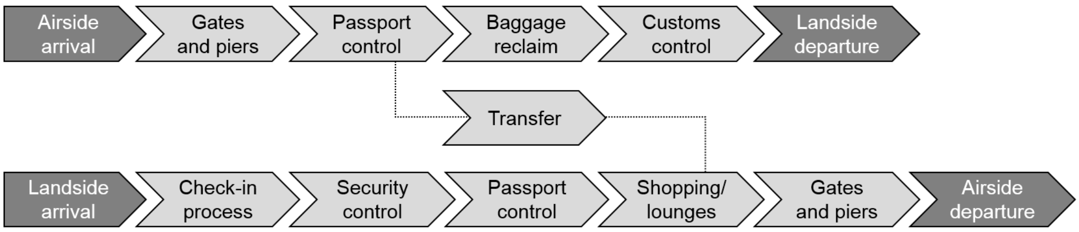


Figure 2.1: Arriving and departing passengers, derived from Gatersleben and van der Weij (1999).

As was seen in the FGAP, passengers can be classified as arriving, departing or transferring. In this thesis, arriving passengers are referred to passengers following the upper flow in fig. 2.1, whilst departing passengers follow the bottom flow. Most of the arriving passengers do not spend time shopping at the airport, whereas departing passengers do. Transferring passengers will de-board their initial flight and go towards the passport control points. After they passed the passport check they continue towards their connecting gates or lounges and wait for their connecting flight. Note that the presence of transferring passengers is dependent on the type of airport, because they are present at hub-airports and not at smaller regional airports.

Furthermore, the distinction can be made between domestic and international flights, or Schengen and non-Schengen , respectively. If the passenger for example is flying international, passports need to be checked (with the exception of EU). In case of domestic passengers, the passport control block could be left out from fig. 2.1.

Passengers can have different purposes of travel, such as leisure or business, and this also affects their behaviour at airports. A leisure passenger tends to go shopping, eating and drinking more often and mostly arrives earlier at the airport. Whereas business passengers tend to arrive later at airports with respect to leisure passengers and spend less time shopping.

Besides the mentioned passenger processes, the terminal lay-out or configuration also affects the time it takes for passengers to arrive at their gate or leave the airport. For example if the terminal has a linear configuration, passengers entering and leaving the airport have short walking distances. However, transferring passengers experience longer walking distances to their connecting flights with respect to other terminal configurations.

Finally, a lot of airports are struggling with the increase in passenger traffic resulting in significant congestion (IATA and ACI (2014)). This presents a problem as an airport wants to achieve an as high as possible level of service (LoS) to their customers (passengers). The level of service indicates to what extend the airport processes offer satisfactory passenger waiting times and adequate space. Level of service is a means of comparing airports and the standards have been defined by the International Air Transport Association (IATA). The key performance indicators (KPIs) that can be used for comparison (Airports Council International (2014)) could be:

- Waiting times or processing rates at airport services,

- Area per person,
- Commuting distances and maximum wait times at transfer processes to ensure minimum connection times.

IATA defined for the levels of service four categories: Over-design, optimum, sub-optimum, and under-provided. A schematic classification matrix is shown in fig. 2.2

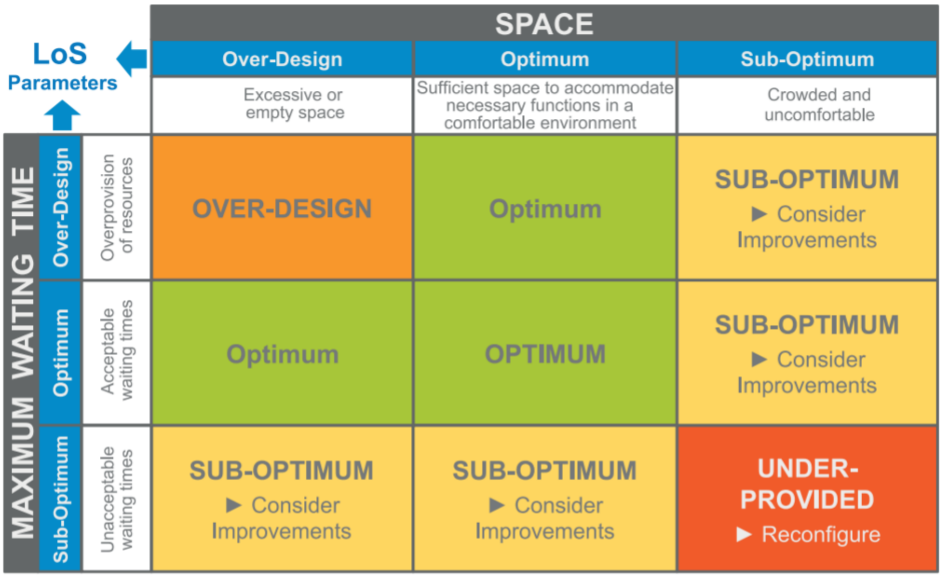


Figure 2.2: Level of service matrix as defined by IATA, taken from IATA and ACI (2014)

The specific values of the LoS parameters have not been listed, but can be found in IATA and ACI (2014).

2.2.2. Agent-based modelling and simulation

Agent-based modelling and simulation (ABMS) is a promising way to model the passenger flows at airports. ABMS is often used to study complex systems and could be seen as part of the artificial intelligence research field. In this section, the general working of agent-based modelling and simulation is described.

"In agent-based modelling, a system is modelled as a set of autonomous decision-making entities called agents"(Bonabeau (2002)). An agent is an autonomous, computational entity that perceives its environment through its sensors and acts upon its environment through its effectors (Weiss (2001)). The agent's autonomy is only constrained by the fact that it is programmed by a human being. So, in this context it means that the agents pursue their goals in an open-ended manner (O'Sullivan and Haklay (2000)). The autonomy of agents gives the agent (to some extent) control over their own behaviour, they can act without the intervention of humans and other systems.

Agents may be affected by other agents, which can be modelled using a multi-agent system. In a multi-agent system, agents interact with each other in an environment to solve problems, achieve goals or execute tasks that are too difficult for one agent to solve autonomously. In order for agents to follow their purpose, a modeller programs agents with behaviours, often described by simple rules, and interactions with other agents, which in turn influence their behaviours.

This brings one to the major benefit of agent-based modelling which is the ability to represent socio-technical factors of agents by making use of a bottom-up modelling approach. This approach aims to identify emerging patterns at the overall group level, by creating agents with specific capabilities at local level. Furthermore, cognitive and social models can be implemented, resulting in diverse agents who are able to make decisions autonomously.

By modelling an entire group of agents on an individual basis, the full effect of the diversity that exists among agents in their attributes and behaviours can be observed and give rise to the behaviour of the system as a whole. A specification of a multi-agent system comprises specifications of the environment, local agent properties (cognitive and/or behavioural), and interaction among agents, and between the agents and the environment.

Agent-based modelling has several benefits over other modelling techniques. The arguments Bonabeau uses are that agent-based modelling:

- **Captures emergent phenomena**: which results from the interactions of individual entities.
- **Provides a natural description of a system**: The model is composed of "behavioural" entities, which makes the simulation of the model seem closer to reality.
- **Flexible**: in the case of an agent-based model means that it is easy to extend and/or tune for example the agent's behaviour. In addition, it is possible to change the level of aggregation of agents.

The environment can be classified by its properties, according to Russell et al. (1995). These are for example:

- **Deterministic / Non-deterministic**: In a deterministic environment an agent's action has a single guaranteed effect, whereas in a non-deterministic environment there is uncertainty about the state that will result from the action.
- **Static / Dynamic**: A static environment remains unchanged over time, except when there are actions performed by the agent. In a dynamic environment other processes are operating on it and change the environment beyond the agent's control.
- **Accessible / Inaccessible**: An accessible environment is one in which the agent can obtain complete, accurate, up-to-date information about the environment's state.

ABMS takes a bottom-up approach, which means that it is built up agent-by-agent and interaction by interaction. There are several types of agents:

- **Reactive agents**: are agents that are able to perceive their environment, and respond in a timely fashion to changes that occur in it.
- **Proactive agents**: are driven not only by observations, but also by internal states.

In the case of a reactive agent, an observation is the input of the agent which directly (or with a delay) results in a response. This type of agent has a set of actions for all possible observations in order to satisfy their design objective. Proactive agents posses an internal/cognitive model and show goal directed behaviour. This internal model could be aimed at solving a problem, planning, decision making, or learning processes. However, the agent does not have complete information of the system. The agents have incomplete information about their environment and are restricted in their capabilities.

Interaction among agents depends on the social abilities that the agents possess, which are important in the formulation of the agent-models. Do they have the ability to communicate in an indirect (through the environment) or direct way? Are they able to coordinate with the aim to achieve (/avoid) desirable (/undesirable) states? Are the agents able to understand and reason about the behaviour and internal states of other agents? Furthermore, one of the choices the programmer needs to make whether or not there is a cooperative setting or a competitive setting.

Agent-based models can be applied and could be useful in any system where emergent phenomena occur. These include social, political and economic sciences. The main areas where agent-based modelling is applied are on studying markets (e.g. stock market), organisations (e.g. operational risk), diffusion (e.g. diffusion of innovation) and flows (e.g. evacuation or customer flow management). The latter area is of special interest, since passenger flow management at airports has become more important.

Concluding, in all agent-based models three aspects should be present such that one could enjoy all the benefits of agent-based modelling:

- **Interaction among agents**; The agents should be able to interact in the environment,
- **Local autonomy**; Agents posses the ability to make decisions autonomously, based on e.g. observations,
- **Diversity**; There are different types of agents present in the model.

2.2.3. Passenger queuing models

One of the underlying passenger processes that has not yet been mentioned, is the process of queuing. Queueing can be viewed as one of the most important parts to consider in the airport environment, as this is where the passengers spend a large part of their time. However, up till now little research has been done in the application of ABMS on queuing systems. Therefore, this side track will give the necessary background knowledge from queuing theory.

The fundamentals of queuing theory were laid by the Danish mathematician A. K. Erlang in 1909 and was taken further by the Russian mathematician A. N. Kolmogorov. Nowadays queuing theory belongs to the classic part of logistics (Šeda et al. (2011)).

In queuing systems, passengers enter the system and require servicing. However, the serving options may be restricted by the amount of service lines. In addition, the service time is random in nature, since it is affected by for example the characteristics of the passenger. Then if all service lines or service desks are occupied, the passenger needs to wait in line to be served. This example, describes queuing system based on a FIFO principle, *First In, First Out*.

In queuing theory it is assumed that arrivals correspond to the Poisson process and that the service time has an exponential distribution Šeda et al. (2016). One of the ways to classify the queues is the A/B/C classification, where

- **A**: the probability distribution of random variable period (interval) between the requirement arrivals to the system,
- **B**: the probability distribution of random variable service time of a requirement, and
- **C**: the number of parallel service lines.

As mentioned, most queuing systems assume the input flow is a Poisson process which has three properties:

1. **Stationarity (homogeneity over time)**: the number of events in equally long time intervals is constant
2. **Regularity**: the probability of more than one event occurring in a sufficiently small interval of length Δt can be neglected
3. **Independence of increases**: the number of events that occur in one time interval are independent of events occurring in other time intervals.

The queuing system's behaviour is represented or described by the Markov Processes in the paper by Šeda et al. (2016). Furthermore, this paper studies the queuing processes by Monte Carlo simulations, which generates the population with a certain probability distribution. One of the key conditions mentioned in this paper is the necessary and sufficient condition for a queue not to grow beyond all bounds. This condition is shown in eq. (2.4), which is the utilisation equation.

$$\frac{\lambda}{c\mu} < 1 \tag{2.4}$$

where λ is equal to the Poisson parameter of passengers arriving, c the number of parallel service lines and μ is the exponential distribution parameter of the service rate.

In an agent-based model passengers could arrive with a certain distribution. For example, most of the leisure people arrive around two hours before their flight, but there are also passengers who arrive earlier or later. Hence, there exists a arrival distribution (or inter-arrival distribution). Kingman (1963) wrote that the waiting time in a queue where the inter-arrival time and service rates have any given distribution (G/G/c queueing system), can be approximated by the following formula:

$$W_q^{G/G/c} \approx W_q^{M/M/c} \frac{C_a^2 + C_s^2}{2}, \tag{2.5}$$

where $W_q^{M/M/c}$ denotes the queue time if the inter-arrival time and service time were both exponentially distributed and, C_a^2 and C_s^2 are the coefficients of variation of service time and inter-arrival time, respectively. Further details on the calculations of the parameters can be found in Kingman (1963).

The limitation of this equation is the fact that it does not perform well when there are no a clear distribution in both the inter-arrival times as well as the service times. In the agent-based model this is the case, as many passenger arrival streams overlap each other. This will result in a non-clear inter-arrival distribution. Furthermore, these equations can be used to approximate the average queue time. However, during the day these queue times can drastically change within a couple minutes. It is especially these peaks that cause disorder and therefore new means should be sought for to predict the queueing time behaviour at an airport.

Agent-based models is a more precise way to model queues. The ABMS is able to show the queue build up real-time and therefore forms the perfect starting-point to look for new ways to model queue behaviour.

2.2.4. Air transport applications

Agent-based modelling techniques have been applied in many research fields like transportation systems (Chen (2010)). Air transport operations are ideal for multi-agent system modelling due to the geographical and functional distribution, and the highly dynamic nature (Burmeister et al. (1997)). Below, three important papers on the application of agent-based models on airport passenger modelling are discussed. The main purpose is to gain information on how passenger processes on airports have been modelled so far using ABMS.

Eilon and Mathewson (1973) are one of the first to propose an ABMS to simulate and evaluate passenger processing times and congestion in the airport terminal. The model presented included a large set of parameters such as flight schedules, passenger characteristics, processing rates at service desks, and the availability of resources.

In the work by Ma (2013), a model of an entire airport was presented, including all processes that the agents (passengers) encounter. The agent would arrive at the airport, possessing several advanced traits such as the ability to make a phone call. Furthermore, the time it takes to make a phone call and the frequency of phone calls were also included. Other traits are for example visiting the restroom, eating or drinking. These advanced traits of passengers were aimed at representing the causal relationships between routing decisions and self-consciousness of passengers. After the initialisation of the passengers, they would go their own ways in the airport and decide on where to go themselves using a mechanism of route-choice decision-making. Passengers would arrive in a range from two hours to 30 minutes in advance of their flight and pass all processes as were described in section 2.2.1. Furthermore, the study also assigned basic traits to the agents. These basic traits included for example the agent's nationality, age, gender, travel class and frequency of travel. A Bayesian networks framework linked the basic to the advanced traits of passengers and was able to infer the probabilities of actions.

In addition, Schultz and Fricke (2011) presented a model which was based on a stochastic approach for passenger movement. In this model the agents posses an operational behaviour level, tactical behaviour level and a safety and emergency planning level. The aim of this research was to create a valid and calibrated agent-based model, since the entire progress of terminal management depends on the individual behaviour of the passengers. This model could then be used for system performance evaluations and for the identification of optimisation capabilities. Included in the model were also the handling processes of the airport such as the security checkpoints.

Another important paper, studies the use of agent-based modelling for security risk assessment. Janssen and Sharpanskykh (2017) argue that traditional methods used for security risk assessment, that are based on probabilistic tools and informal expert judgements, lack the capability to take the dynamic and intelligent nature of attackers into account. Therefore, Janssen and Sharpanskykh propose a combination of agent-based modelling and Monte Carlo simulations. An illustrative case study is given of an terrorist that aims to bring in an improvised explosive device (IED) past an airport security checkpoint. Three types of agents are included in the model: An attacker, a defender and passengers. Furthermore, the environment in which the agents interact consists of sensors (e.g. X-Ray machine) and physical objects (Walls and queue separators). The queue separators allow for the measurement of the number of people in the queue and the average queuing time.

As mentioned earlier, agent-based modelling and simulation is also used to model evacuation processes. Cheng et al. (2014) consider the passenger group dynamics on an airport evacuation process using an agent-based model. Results showed how passengers react to an evacuation signal, which route to choose in an evacuation and the average time for passengers to finish the evacuation. This work shows that an agent-based model is well suited to simulate the evacuation process at an airport and analyse the pedestrian group dynamics. Airport managers could use the simulations to propose and test evacuation plans.

There are also other simulation studies related to passenger flows in airport terminals. An example is the paper by Gatersleben and van der Weij (1999), where passenger flow management is studied at Amsterdam Airport Schiphol. They apply simulation to gain insights into the relations between processes, the presence of bottlenecks and their causes. Since the simulation provided continuous output, time-dependent graphs could be constructed of utilisation, throughput and waiting times. Furthermore, the variance of occupation of the terminal segments could be analysed. Several scenarios were simulated and by comparing the results the likelihood of arising bottlenecks could be estimated. The study proved to be very successful.

2.2.5. Validation of agent-based models

There are many advantages of using agent-based modelling and simulation in complex systems. However, one of the main draw-backs is the ability to validate the model. Because agent-based modelling is a way to describe what is happening in the real world, it is necessary to validate that the model is actually describing the real world. The credibility of the model depends highly on the ability to validate the model.

Often, there is a trade-off between increasing the confidence in the level of accuracy of the model, and the cost of data collection and the effort required to validate the models. The validation of a pedestrian flow agent-based model is difficult, due to the amount of parameters that are included in an agent-based model (Teknomo and Gerilla (2005)). One way to collect validation data could be by using video cameras at the specific locations. Using the videos, certain aspects can be validated such as speed of overall flow and instantaneous occupancy by humans at the considered location. In addition, tracking software could be used to show the paths passengers take around the airport terminal. Besides, validation of pedestrian flow models require a deep understanding of the factor and parameter behaviour.

Simulation models can be validated using mathematical statistics as described in the paper by Kleijen (1999). Three situations can be distinguished in which statistical validation techniques might be necessary. When there exists:

1. no data on the input or output of the real system,
2. only output data on real systems,
3. both input (or trace) and output data on real systems, which is used to perform so-called trace driven or correlated inspection simulation.

In case (item 1), an analyst could still generate simulated data and perform sensitivity analysis to test whether the simulation model contradicts qualitative, expert knowledge.
In case (item 2), the means of real and simulated output distributions may be compared using the two-sample student t test.
Finally, in case (item 3) alternative regression and bootstrap procedures should be applied. The alternative regression should be applied if the outputs are normally, identically and independently distributed

(n.i.i.d.), and is a regression of the differences on the sums. Alternatively, one could use bootstrapping of a simple validation statistic based on differences.

2.3. Simulation optimisation

In this section, the most relevant theoretical content is discussed: Simulation optimisation. This research field has the potential to close the existing scientific gap.

In mathematical programming models a large number of decision variables and constraints can be involved. Similarly, simulation models often utilise a large number of random variates. However, the combination of the two most used tools on this scale is not yet achievable (Fu et al. (2015)). Mostly due to the fact that the dynamics of simulation models cannot be simply converted into e.g. a set of constraints. Hence, this section studies one of the attempts to close this scientific gap. Simulation optimisation is the process of finding the best values from among all possibilities for the input variables without evaluating each possibility. The aim is to maximise information obtained with the lowest amount of resources (e.g. time).

A simple flow chart of a simulation optimisation model is shown in fig. 2.3.

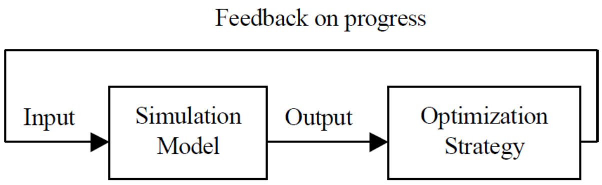


Figure 2.3: Schematic overview of a simulation optimisation model, taken from Carson and Maria (1997).

The simulation model has n input variables and m output variables. Hence, simulation optimisation tries to find the optimal settings of input variables such that the output variables of interest are optimised.

In simulation optimisation there has to be a trade-off between allocating computational resources used for searching the solution space versus running additional simulation replications to increase the estimating performance of promising solutions. The optimisation process involves algorithmic computation, as well as simulation computation for estimating the new candidate solutions. Therefore, two types of sampling are needed: Sampling of the solution space and the sample path (stochastic simulation) space.

A general simulation optimisation problem could be of the form as displayed in eq. (2.6). In this example, there is only one output variable of interest (y).

$$
\begin{array}{lrcl}
\min\limits_{\theta \in \Theta} & y(\vec{\theta}) \equiv E[L(\vec{\theta}, \omega)], & & \\
\text{s.t.} & a(\vec{\theta}) & \leq & b, \\
 & c(L(\vec{\theta}, \omega)) & \leq & d,
\end{array}
\tag{2.6}
$$

where $\vec{\theta}$ is the n-dimensional vector of all the decision variables and Θ is the feasible region. There is little knowledge on the structure or shape of y in eq. (2.6), such as linearity or convexity. The variable y cannot be obtained directly. However, it is an expectation of another function $L(\vec{\theta}, \omega)$, which is a sample performance estimate obtained from the output of the simulation replication. This could for example be the queue-time for a certain amount of passengers for a specific gate assignment plan. Furthermore, ω comprises the randomness in the system.

There are six major categories of simulation optimisation methods as described by Carson and Maria (1997). The categories can be found in fig. 2.4, together with some of the most used techniques employed in the different methods.

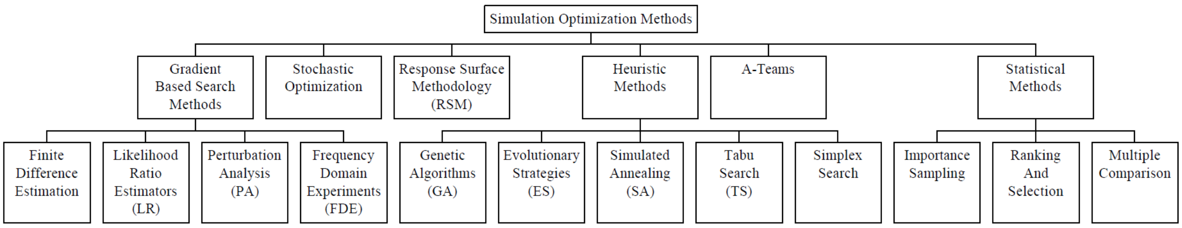


Figure 2.4: Simulation optimisation methods as given by Carson and Maria (1997).

The most used categories in simulation optimisation methods are briefly explained below, for informative purposes only.

- **Gradient based search methods**
  This methods assesses the objective function's shape by estimating the response function gradient (∇y). Deterministic mathematical programming techniques are then used to solve the problem.

- **Stochastic optimisation**
  These methods are iterative algorithms based on gradient estimation. In stochastic optimisation, local optimum have to be found for an objective function whose values are not known analytically. However, the values can be estimated or measured. A disadvantage is that when the objective function is flat, these algorithms tend to converge very slow and often diverge when the objective function is steep. The counterpart of stochastic optimisation is sample path optimisation. In this method a stochastic problem is turned back into a deterministic problem, by taking a large enough set of samples. The problem is then optimised using a non-linear programming method.

- **Response surface methodology**
  The response surface methodology (RSM) is explained in Kleijnen (2008). It is a procedure for fitting a series of regression models to the output variable and optimising the resulting regression function. An application of RSM in simulation optimisation is shown in Biles (1974).

- **Heuristic methods**
  These are direct search methods that require only function values. This category includes approaches such as genetic algorithms, tabu and scatter search approaches and any other iterative and possibly population-based (evolutionary) algorithms from deterministic optimisation.

The interest in the area of simulation optimisation is growing. However, many of the methods require a rapid determination of the output variable value at a given input parameter vector value. Therefore, recently much attention is given to shorten the computational time of these methods by means of meta-modelling.

2.3.1. Meta-modelling

The agent-based model of the passengers moving around an airport is too large and complex to include it in any other model, or to use it for smaller analyses. Therefore, a way should be found to reduce the size of the agent-based model by approximation, whilst not losing a certain level of detail in the model. Such a model, is called a meta-model for simulation input-output relations. A simple form of a meta-model can reveal the general behaviour of the more complex agent-based model. These simpler form of models can be run iteratively for repeated *what-if* evaluation for multi-objective systems or for design optimisation (Barton (1992)). The advantage of a meta-model based optimisation strategy is the incorporation of knowledge of the smoothly varying response function and it enables a reduction in prediction variance by extending the effect of the law of large numbers over all fitting points. However, one should be careful because bias is introduced when the meta-model fails to capture the true nature of the response function.

The general mathematical form of a simulation model input-output function will be represented as:

$$\vec{y} = \vec{g}(\vec{v}) \tag{2.7}$$

following the paper by Barton (1992). In this simple representation, $\mathbf{y}, \mathbf{v}$ represent the output and input vectors, respectively. Meta-models are most often built per output component of $\mathbf{y}$ (each coordinate function of $\mathbf{g}$). In the work by Barton (1992), the attention is restricted to input-output models where the random component (if present) is additive, $\mathbf{y}$ has one component, and the list of parameters is restricted to those that will be in the argument list of the meta-model:

$$y = g(\mathbf{x}) + \epsilon. \tag{2.8}$$

Creating a meta-model involves finding a way to model g and ϵ. In analogy to the work by Barton, the meta-model is denoted by f and the predicted output responses as $f(\vec{x})$ or $\hat{y}$. Hence,

$$g(\mathbf{x}) \approx f(\mathbf{x}) = \hat{y}. \tag{2.9}$$

In mathematical form the meta-modelling work does not seem too difficult. However, several issues will be encountered when building a meta-model:

- Choice of functional form,
- Design of the experiments, which sets of $\mathbf{x}$ with observations y to fit f to g, the assignment of random number streams, the length of runs etc.,
- Assessment of adequacy of the fitted meta-model.

A meta-model inherently loses details, which were present in the parent model (ABM). The loss of detail is determined by the level of abstraction. In table 2.1 a general categorisation is given of meta-models.

Table 2.1: General categorisation of meta-model types as given in Manfren et al. (2013).

Type	Description	Advantages	Disadvantages
White-box	Models with high level of detail based on the laws of physics that permit accurate modelling, employing algebraic and differential equations (ODE, PDE) to describe temporal and spatial variations.	High level of accuracy and precision. Gives a detailed physical description of phenomena.	High computational effort. Difficult and error-prone modelling and implementation process.
Grey-box	Model uses a simplified description of the underlying phenomena in space and time with algebraic equations and first order ODE. Data is used to identify the best model and corresponding model parameters.	Easier to implement compared to white-box models. Gives a physical description of phenomena. Good level of computational efficiency.	Less accurate and precise than white-box models. Error-prone implementation process.
Black-box	Empirical or data-driven models. Based on little or even no physical behaviour of the system. Only rely on the available data to identify the model structure.	High level of computational efficiency and flexibility. Simple to implement with respect to the achievable accuracy.	Physical representation is gone. Opaque to the user.

So in general, a simplified physically based model with unknown parameters optimised with respect to real data (grey-box model), is more understandable than a black-box model. However, a black-box model is fairly simple to implement and could achieve high accuracy. Furthermore, the computational efficiency is very high of black box models, making it possible to do multiple runs in less time than a grey-box model could. Hereafter, two meta-models will be treated which are often used in simulation optimisation methods as stated by Kleijnen (2008).

Polynomial regression

The functional form of a meta-model will generally be described as a linear combination of basis functions from a parametric family (eg. polynomials, sine functions etc.). The most popular techniques have been based on parametric polynomial response surface approximation: The function g can be approximated in some region of the x's by a polynomial model. These polynomial regression models were developed for optimisation (Barton (1992)), as explained earlier.

A meta-model is built using m simulation outputs $\vec{y} = (y_1, ..., y_m)'$, which were a result of the input conditions $\vec{x}_1$, ..., $\vec{x}_m$. The errors (ϵ_i) for the different observations i are assumed to be independent, identically distributed Gaussian quantities with variance σ^2. A general form of a polynomial regression model is given in eq. (2.10).

$$f(\vec{x}) = \sum \beta_k p_k(\vec{x}) \tag{2.10}$$

The function $p_k(\vec{x})$ represents a basic function and are usually taken as the products of power functions, 1, x_j, x_j^2,.... The β_k are estimated using the observed m data points using the least squares method of maximum likelihood estimation.

The interpretation of the models are also dependent on the polynomial chosen for the regression. For example, a β of a linear term x_1, indicates the direct relationship with the outcome y. If the value of the β coefficient is large, then the parameter has a large effect on the outcome and vice versa. A coefficient of a quadratic term indicates whether or not there is a non-linear response. Furthermore, a large coefficient corresponding to a cross product term (e.g. $x_1 x_2$) is interpreted as a change in the effect of x_1 as a function of the value of x_2, and vice versa. In addition, calculating the estimates of the coefficients β is relatively simple and for low order polynomial fits the accuracy of the predicted value does not rapidly degrade when moving away from any experimental observation. “Although polynomial response is not accurate for highly non-linear problems, it is easy to use and is very accurate for low-order non-linearity” Jin et al. (2001).

Kriging meta-model or spatial correlation model

An alternative meta-model is the Kriging meta-model (KG), or sometimes called spatial correlation model. Kriging models are fitted to data that are obtained for larger experimental areas than the areas used in low-order polynomial regression meta-models. Hence, these models are global instead of local models.

This method captures the expected smoothness of the function in a spatial correlation function. The model assumption is:

$$y(\mathbf{x}) = g(\mathbf{x}) + Z(\mathbf{x}). \tag{2.11}$$

In eq. (2.11) Z is assumed to be a realisation of a stochastic process with mean zero (Gaussian process) with a spatial correlation function as in eq. (2.12).

$$Cov(Z(x_i), Z(x_j)) = \sigma^2 R(x_i, x_j) = exp(-\sum \theta_j (x_i - x_j)^p) \tag{2.12}$$

where σ^2 is the process variance, and R is the correlation. As mentioned in work by Barton (1992) the value of p is fixed at $p = 2$ such that it is a Gaussian correlation function. Furthermore, $g(\vec{x})$ is usually approximated by a constant, or a linear function of x. Maximum likelihood is used to estimate the θ_j values, which indicates the importance of input j. The higher θ_j the less effect input j has. The θ_j values are used to calculate approximate expected values of eq. (2.11) to provide the meta-model $f(\vec{x})$. Mentioned in the book by Kleijnen is that a Kriging meta-model can be used to model deterministic simulation models as well as stochastic simulation models. The latter is done in van Beers and Kleijnen (2003), where the stochastic simulation output is simply replaced by the average computed from the replications. Another paper which clearly describes the global optimisation of a stochastic black-box system using the Kriging meta-model is the work by Huang et al. (2006).

In the work by Kleijnen (2009) a more elaborate overview of basic Kriging assumptions and formulas is given and a comparison with classic linear regression models is made.

Radial basis function

The radial basis function approximation consists of a sum of radially symmetric functions centered at different points in the domain Θ. The radial basis function model is special type of neural network consisting of three layers. The first distributes the input vector to each of the receptive field units in the second layer, which is the hidden layer. The hidden units play a role in simultaneously receiving the input vector and non linearly transforming the input vector into an m-dimensional vector. The outputs of the m-hidden layers are then linearly combined with weights to produce the network output at the output layer. The radial basis function model can be described as

$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j \varphi_j(\mathbf{x}) = \sum_{j=1}^{m} w_j \varphi(\|\mathbf{x} - \mu_j\|/\sigma_j), \tag{2.13}$$

where $\vec{x}$ is the input vector, μ_j is the *j*th basis function centre, $\|\cdot\|$ denotes the Euclidean distance, the w_j values are the weights, and the σ_j are basis function widths. The $\varphi()$ plays the role of transfer function.

This type of meta-model has proven to applicable to a M/M/1 queuing system (Miyoung Shin et al. (2014)). In this paper the function (φ) was taken as a Gaussian basis function, which is one of the most popular forms. The input parameter taken for the meta-model was the arrival rate. The Gaussian radial basis function model is used to estimate the sojourn time in the system (waiting time plus service time).

Similar to the paper by Miyoung Shin et al. (2014), other reference papers (such as Miyoung and Goel (2000)) that use a Gaussian radial basis function also only have one input variable. The aim of current study is to approximate the agent-based model and simulation by a set of meta-models, which are capable of capturing the dynamic relationships from the agent-based model (ABM). Therefore, it is believed that only one input variable to describe one output variable will not be enough.

All three meta-models presented methods show great potential. However, up till now little (to no) research has been done on applying these meta-models to input-output relations of agent-based models and simulations.

2.4. Literature conclusions

This chapter has provided a broad literature review which has uncovered the scientific gap that is the connection between operations research optimisations (FGAP) and airport passenger flow modelling (ABMS). It has been found that simulation optimisation forms the right basis to close this scientific gap. In this section the main conclusions from this research are summarised.

2.4.1. Flight-to-gate assignment problems

Flight-to-gate assignment problems have been assessed for decades and the present papers apply only minor changes to models proposed in existing literature. It can be concluded that the deterministic models, do not fit the dynamic character of an airport. Furthermore, the majority of the literature using robust models took a one-sided approach by making the assignment plan robust for flight delays. The scientific gap that can be defined in the FGAP research is probably the ability to capture the real world environment, especially on the terminal side. Unforeseen events do not only occur on the airside of the airport. Disturbances on the terminal side, like bottlenecks in passenger flows or queue times at security checkpoints, could be reason for an airport operator to intervene in the assignment plan. Unfortunately, little attention is given in existing research to the terminal side processes that might have an effect on the flight-to-gate assignment. Finally, three solution methods have been discussed: Expert methods, Simulation methods, and Mathematical programming techniques. The majority of the literature reviewed, uses the mathematical programming techniques. However, if the computational time was critical or the objective function was non linear, one could switch to meta-heuristic methods. It has been proven in literature that the differential evolution algorithm was able to find even global optimum faster than other global optimisation algorithms. In addition, there are no restrictions to the objective function or the constraints which makes the DE algorithm an ideal starting point.

2.4.2. Passenger flow modelling at airports

This section gave a clear overview of the processes that a passenger encounters in an airport. Arriving, departing and transferring passengers pass different areas in the airport. The airports are designed to provide a certain level of service. However, due to the continuous high growth in passenger traffic these levels of service standards are endangered. Agent-based modelling and simulation has been proven to be very successful in modelling passenger flows at airports. The passengers can be modelled using a multi-agent system, where the agents interact in an environment such as an airport. Analysing the results of a multi-agent simulation can be very useful to identify bottlenecks at the airport. Furthermore, agent-based modelling and simulation is able to show the emergent phenomena during the day due to its bottom-up approach. The difficulty with agent-based models however, is the fact that validation process is difficult. Often, available data is little which calls the need for experts with a deep understanding of the passenger's behaviour.

2.4.3. Simulation optimisation

Simulation optimisation will form the theoretical content of this thesis. Simulation optimisation could be seen as optimising an expectation function with stochastic parameters. The aim is to find optimal settings of input variables such that the output variables of interest are optimised. The most often used methods were mentioned and briefly explained. One of the major difficulties in simulation optimisation is the time needed to simulate a feasible solution to the optimisation problem. Therefore, simulation model input-output meta-models were examined, which reduce the size of the simulation model by simplification. Polynomial regression, Kriging and radial basis function meta-models were briefly explained, as these are the most often used meta-models. The latter two have been proven to be promising in representing a M/M/1 queuing system. Simulation optimisation describes both direct and indirect (using meta-modelling) methods to integrate simulation models with optimisation models. Hence, simulation optimisation forms the perfect theoretical basis on which the current study can build.

3

Research framework and methodology

As was described in the introduction, this research will focus on the integration of agent-based model and simulation with an flight-to-gate assignment optimisation. Before one can start integrating these two paradigms, a clear path should be specified such that the most is achieved during the limited time of the research.

The first step is to identify the problem which will be solved, this is done in section 3.1. Then, in section 3.2 the objective for this thesis will be presented and research questions are formulated. Every research needs a scope, which is defined in section 3.3. The scope defines the boundaries of the research, which are needed to keep the research goals realistic. Next the model requirements are presented in section 3.4, which explains the expectations for the model and the software used to create it. Furthermore, in section 3.5 the contribution of this research is summarised by making a distinction between scientific and industry contribution. The chapter is concluded by presenting the methodology in section 3.6 that will be used to answer the research questions.

3.1. Problem statement

One of the most important and complex tasks at an airport is the development of the flight-to-gate assignment plan. The gate plans are made at least one-day in advance, but disturbances in the schedule require also real-time adjustments. A vast amount of literature has focussed on solving the flight-to-gate assignment problem. Assigning gates in order to minimise waiting times, connection times, un-gated flights, and baggage transport distances.

The most popular objective found in the flight-to-gate assignment problem papers, was the passenger walking distances (as in the work by Mangoubi and Mathaisel (1985)). However, static optimisations like this one do not fit the dynamic environment of an airport. A passenger walking at an airport is not only confronted with the distances it needs to walk. But the processes it encounters such as the check-in and security processes also affect the passenger experience. Currently in the FGAP research, efforts are being made to make the airside more realistic by including flight delays as in the work by Yan and Tang (2007). In addition, models are being made that are robust to disturbances using robust models. An example is the model proposed by van Schaijk and Visser (2017), which incorporates the airlines' punctualities into the model. However, realistic modelling of the terminal-side is underexposed. One can imagine that congestions at airports could give an incentive to the passenger to choose a different path. Hence, the pre-assumed walking distances/times in the FGAP are not the actual walking distances/times.

Passenger traffic is growing to the extent that airports reach their operating limits. This means that airports are getting more and more congested, possibly leading to airside delays. It was found by Eurocontrol (2017), that 5-12% of the flights delayed due to landside terminal elements was due to the security checkpoint process. In addition, there is a growing concern of terrorist attacks which puts a high pressure on security. Due to the fact that airports are often restricted in building expansion, airports need to come up with smart and sophisticated solutions to overcome congestions at the airport.

Agent-based modelling and simulation has proven to be a suitable way to model and analyse passenger flows at airport terminals. An agent-based model is able to simulate a specific day or scenario at a specific airport and will reveal emergent phenomena. These could include bottlenecks in the terminal building or queue build up at processing points. Several studies examined the whole passenger experience at an airport (Ma (2013)), at specific passenger processing points (Janssen and Sharpanskykh (2017)), or examined the passengers during evacuation of an airport Cheng et al. (2014). Ideally, an agent-based model would be used to find an optimal solution that minimises the occurrence of unwanted bottlenecks or queues. However, due to the amount of details (characteristics) of the individual agents (passengers) and the number of agents in the entire system, the agent-based model could become too big and complex to perform iterative optimisations with.

Another problem is the fact that there is a disconnect between the management of terminal processes and control over airside operations. So, there are a lot of processes, such as the gate assignment, that could change the guidance of passenger flows at airports, but these have not been explored in research yet.

3.2. Research objective and questions

The goal of this research is to fill the scientific gaps mentioned in chapter 2. The proposed research objective is:

To develop a methodology to integrate an agent-based model for airport terminal processes with a flight-to-gate assignment optimisation, such that these airside and landside processes can be managed simultaneously.

By integrating the agent-based model with a flight-to-gate assignment optimisation, a completely novel approach to manage landside and airside simultaneously is presented. To reach the objective, a main research question has been formulated. The objective is achieved once an answer to this question is found. The main research question is the following:

How can a multi agent-based model for airport terminal processes be integrated with a flight-to-gate assignment optimisation?

To answer this question, the research has been split up into four sub-research questions. Once these have been answered, the main research question can be answered. The sub-research questions are:

1. *What airport areas are important to consider when integrating a multi agent-based model for passenger processes in a flight-to-gate assignment problem?*
2. *What variables used in the agent-based model from areas in item 1, could be used in the flight-to-gate assignment model?*
3. *Can the agent-based model be integrated in a direct way or by creating meta-models that are able to capture the dynamic relations?*
4. *Is the integrated model credible, does it reflect the agent-based model and reality?*

It is hypothesised that the integration can take place in both a direct way and indirect ways, as was seen in the simulation optimisation literature. However, there are probably infinitely many ways to integrate the two paradigms and therefore the research scope needs to be defined.

3.3. Research scope

In this section the research scope is presented within which the research is performed. The scope is determined based on the scope used in flight-to-gate assignment problem literature as well as the insights from the AAS case (see chapter 1).

Given the time constraint of this thesis the scope has been restricted to only a small section of a possibly larger airport. The added value comes from developing the integration methodology and not from solving the largest or most complex problem at hand.

Geographically, the research will be limited to one airport. However, the aim of the research is to create a general methodology which can be applied to any airport around the world. The agent-based model that will be used for integration will contain departing and transferring passengers. Hence the focus of the research will be on a hub-airport similar to Amsterdam Airport Schiphol. One of the characteristics of a hub-airport like Schiphol is the fact that there exists a Schengen area and a non-Schengen area. Hence, this will also be used in current research. In addition, a variety of aircraft sizes arrive at hub-airports leading to potentially interesting results or insights.

Landing and taxiing will not be considered in the airside of the model, as most of the FGAP literature did not take this into account as well. Furthermore, the temporal scope of the flight-to-gate assignment that will be created will span over half a day of operations.

Ideally, an airport section will be considered with multiple check-in areas, security checkpoints, border control points, and two wings with gates. Depending on the gate selected, passengers have the possibility to take different paths from the entrance to their assigned gate. This situation could probably have the most interesting results as passengers flow from all directions and are difficult to manage by airport management. The basis on which the fictitious airport will be based are wings C and D of Amsterdam Airport Schiphol. This part of Schiphol contains the above mentioned points.

Finally, the initial aim of the research was to integrate all processes of the agent-based model with a flight-to-gate assignment model. However, this appeared to be unrealistic due to the time constraint. Therefore, it was decided to examine only the security checkpoint queues. The security checkpoints were chosen as these are one of the largest passenger processing points contributing to flight delays (Eurocontrol (2017)). Furthermore, during preliminary exploration it was found that the most interesting and critical behaviour was observed at the security checkpoints.

3.4. Model requirements

To achieve the research objective, models will be built that will help to achieve the research goals. Below a few of the model requirements and the software used will be discussed.

The model requirements are:

- The flight schedule input for the flight-to-gate assignment should be representative of a real flight schedule.
- Strong relationships between gate assignments and simulation outputs will need to be present.
- The agent-based model should be representative for actual daily airport operations.
- The flight-to-gate assignment model should be a simple flight-to-gate assignment model that could be used by airport managers.
- The results should be presented in graphs and tables.
- To be able to substantiate the successfulness of integration, evidence should be provided.

These requirements will be translated to multiple models, which will be built in MATLAB and JAVA. The central modelling and simulation environment in this research is AATOM, which will be elaborated on in section 3.6. AATOM has been coded in the JAVA language and the open-source software development environment used is Eclipse. The models that will be built in MATLAB consist of the meta-models, the optimisation algorithm and the problem definition.

Improving the computational time of the optimisation algorithm and agent-based model will be outside the scope of this research. The computers used to run simulations, build meta-models and perform the optimisations are HP Elitebook 8560w computers with an Intel Core i7-2630 QM processor. At one point during the research, three of these computers were needed to speed up the process.

3.5. Impact and contribution

The combination of two most used research tools, mathematical programming models and simulation models, is in general a gap in the current literature. Therefore, it is important to explore this gap, in addition to the gap between the airside and terminal processes.

The novelty of this research is the fact that the implementation of a dynamic passenger model (such as an agent-based model) in a flight-to-gate assignment problem, has never been done before. This could be a major improvement of the current way passengers are taken into account in the flight-to-gate assignments. This combination could potentially create a model that is capable of handling/optimising airside as well as landside processes simultaneously.

This research aims to open a whole new direction in both operations research as well as the agent-based modelling field. As discussed, the combination possibilities of both paradigms are endless, and this research focusses only on one specific case.

The literature reviewed in chapter 2 identified gaps in the existing literature. Figure 3.1 presents an overview of the state-of-the-art and potential contribution of the proposed research from both a scientific and industry perspective.

	Scientific	Industry
Status quo	• Robust flight-to-gate assignments, accounting for passengers by e.g. distances. • Agent-based models for terminal passengers.	• Passenger experience has only limited impact on gate assignment. • Agent-based models used to test scenarios.
Contribution	• Break paradigm ‘walls’, open a new direction of research possibilities. • Represent passengers as dynamic factor in static FGAPs. • An initial methodology to integrate an ABMS with OR.	• Show a way to integrate airside with landside processes. • Define a method to optimise the use of facilities at the airport.

Figure 3.1: Overview of the status quo and contribution of the proposed research, from both scientific and industry perspective.

3.6. Methodology

Having set the objective and defined the boundaries of the research, the next step is to define the methodology that will be used to achieve the research objective.

3.6.1. AATOM and simulation environment

Central in this thesis is the agent-based modelling and simulation environment, AATOM. AATOM stands for Agent-based Airport Terminal Operations Model. It is a microscopic agent-based model that is able to simulate movement and operations in an airport terminal (Janssen et al. (2017)). The creator of AATOM is Stef Janssen, who is currently developing the platform even further. At the beginning of this research the model included the main handling processes required for outbound aircraft. These include check-in and security checkpoints. Furthermore, basic facilities for discretionary activities like bathrooms, restaurants and shops, are modelled. It has been designed in such a way that students can add or remove functionalities and create their airport environment the way they need it. Hence, it forms a basis for studies, to investigate the field of security, efficiency and to investigate the relationship between other parameters. This model is the ideal starting point for the current research.

This thesis is still one of the first theses that uses AATOM as an experimental environment. Therefore, some of the features that were not yet present when starting the research had to be developed during the thesis as well.

As the objective is to integrate an agent-based model for airport passengers, the first step is to generate an airport environment. As discussed in section 3.3, two wings (C and D) of AAS would form the inspiration for the fictitious airport to be created. This means that the dimensions of the terminal building of AAS should be used to create the environment. Unfortunately, the exact dimensions of in general terminal buildings are confidential (for security reasons), but schematics of the terminal building are often available. The design is therefore based on those schematics, together with knowledge about the gate spacing. A total of four gates are present at the fictitious airport and the terminal building is around $7000m^2$. The length of the piers will be kept to a minimum since these will not add value to the research.

Next, the processes and passengers that are within the scope are added. These include the check-in, security checkpoints, and border control facilities. The placement of the facilities will also be inspired by AAS. The border control facilities will split the airport in half, with on one side the Schengen area and on the other side of the border control the non-Schengen area. A total of four check-in facilities, four security checkpoints and two border control facilities are added. Creating four distinct paths to the four gates.

In the current state of AATOM (when the research started), passenger diversity was restricted to the luggage type passengers are bringing. One type of passenger only brings carry-on luggage whereas the other type of passenger has checked luggage. The autonomous decision making in the current state of AATOM is visible in two ways. The first is the decision people make whether they would like to check-in at the airport or prefer to check-in online. Logically, people who checked-in online will not have to pass the check-in facility, provided they do not bring checked luggage. The second autonomous decision the agents make is whether they visit some of the bathrooms, restaurants and shops. It was discussed in section 2.2.2 that the local autonomy, diversity and interaction among agents would create the most valuable agent-based model. In addition, the scope of the research is on a hub-type of airport where also transferring passengers are present. Therefore, it was decided to enrich the diversity among agents and add another autonomous decision that the agents could make. The diversity was enriched by creating two types of passengers with different purposes of travel. The first division was between departing and transferring passengers. Secondly, the purpose of travel could be leisure or business. The diversity of agents posses different characteristics and decision making capabilities. The autonomous decision capabilities were extended by creating the possibility to dwell around the airport between check-in and proceeding towards the security checkpoint. Based on stress build up (growing queues), agents can decide when to proceed. In addition, different arrival distributions are added for business and leisure passengers based on literature.

The next step is to perform Monte-Carlo of many realistic departure schedules, using the pseudo-random number generators used by many processes that are implemented in AATOM. These simulation results are used to size the passenger handling facilities. The data used to generate the schedules have been taken from the FlightRadar24 website. This data was used to asses the sizes of aircraft that depart from AAS every day and the frequency of departures with respect to the total number of gates. Simulating a large amount of different scenarios were useful to size the facilities such that in some scenarios congestions only would occur at the security checkpoints. In reality this means that the facilities (except for the security checkpoints) are slightly over-sized for the most heavy scenario.

3.6.2. Simulation optimisation

The first way to integrate the agent-based model with a flight-to-gate assignment problem is a direct integration by simulation optimisation. In this integration the entire agent-based model will be integrated with an operations research optimisation algorithm. As discussed in chapter 2, the disadvantage of this way of integration is the fact that the optimisation is computationally heavy. The advantage however, is that there is little loss of detail. The simulation optimisation methodology applied to integrate ABMS with the flight-to-gate assignment optimisation is graphically shown in fig. 3.2.

First, the flight-to-gate assignment problem will be specified that will be solved by the optimisation algorithm. Since this paper is pioneering the integration of an ABMS with an FGAP, the problem to be

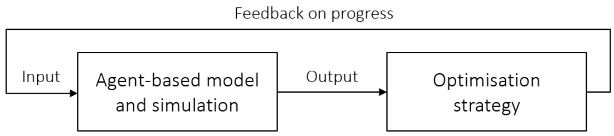


Figure 3.2: Simplified representation of the simulation optimisation method used to integrate an agent-based model with a flight-to-gate assignment optimisation, based on figure from Carson and Maria (1997).

solved is taken as a relatively simple flight-to-gate assignment problem. Once a proper methodology has been created for the integration, one could easily increase the complexity or scale of the problem.

The most difficult part to connect these two paradigms, agent-based modelling and flight-to-gate assignment optimisations, is that up till now there is little connection between the decision variables in a flight-to-gate assignment problem and the behaviour in the agent-based model. In general, passengers enter the airport and proceed to the airline check-in desk with whom they have their flight. Next, they continue towards the security which consists out of multiple checkpoints that mostly serve a single pier or multiple piers with gates. Since passengers with different destinations enter the security queue, there is little (to no) causal relationship between the assignment of gates and the security checkpoint queue behaviour through passenger flows. Therefore, in section 3.6.1 a causal relationship was created by having distinct security checkpoints per gate. Then, if a flight was assigned to a gate, the queue build up would be due to the sequence of flights assigned to that gate.

The starting point for the flight-to-gate assignment problem was the model by Mangoubi and Mathaisel (1985) shown in eq. (2.1), whose objective was to minimise the passenger walking distances. The decision variables used by Mangoubi and Mathaisel will be used in current study as well. In addition, extra decision variables are added to control the number of X-ray scanners that are open during the planning horizon.

The aim of current research was to avoid congestions and enhance the passenger experience at the airport. This means that the queue times passengers observe should be minimised from the perspective of the airport. An individual passenger would like to have zero queue time. However, since there is only a limited amount of resources at the airport, this would mean that passengers with different destinations will experience queue. Therefore, the objective has to be specified such that the optimisation will lead to the best spread in queue times among the different security checkpoints. In the simulation optimisation method, the objective function value will be the result of many replication simulations of one scenario/trial solution. The restrictions posed on the flight-to-gate assignment will be similar to the constraints in eqs. (2.2) and (2.3). In addition, restrictions will be posed on flight with Schengen and non-Schengen destinations and on the amount of resources that can be deployed.

Finally, the flight-to-gate assignment is optimised using meta-heuristics as these algorithms are able to handle non linear formulations of the flight-to-gate assignment problem. Furthermore, the literature review has revealed that the differential evolution algorithm was able to outperform other global optimisation algorithms. Even in the case of a discrete solution space.

The way the actual optimisation will take place is discussed in chapter 5. One of the major disadvantages of this direct integration will be the computational time to find an optimum for the flight-to-gate assignment problem. Therefore, indirect ways should be developed that replace the agent-based model in the optimisation strategy by an abstract representation.

3.6.3. Meta-modelling

The abstraction process is called meta-modelling for simulation input-output relations. During this process, equations are generated which should be able to replace the agent-based model in the optimisation strategy.

Therefore, the output of the models should be related to the earlier defined objective function and the input variables should be related to the decision variables. In this thesis one specific way will be chosen to connect the decision variables of an FGAP to the response variable of the meta-models. Since the assignment of a flight to a gate will initiate the arrival of passengers during a period of time. Hence, first an assumption should be made on the arrival distributions of passengers (at entrance) as well as the known amount of passengers that will depart with a certain flight. Next, with these known incoming streams of passengers, the meta-models will try to predict the security checkpoint queue times. Hence, the connection between the decision variables of the FGAP with the objective function is indirect. In the intermediate step the decision variables are converted into passenger arrival streams after which meta-models will predict the queue times per security checkpoint. The resource decision variable (which will select the number of X-ray scanners used) will select which set of meta-models will be used to predict the queue times.

The above described method is graphically shown in fig. 3.3. The meta-model sets in the figure indicate the different meta-models of a single type constructed for different amount of resources deployed.

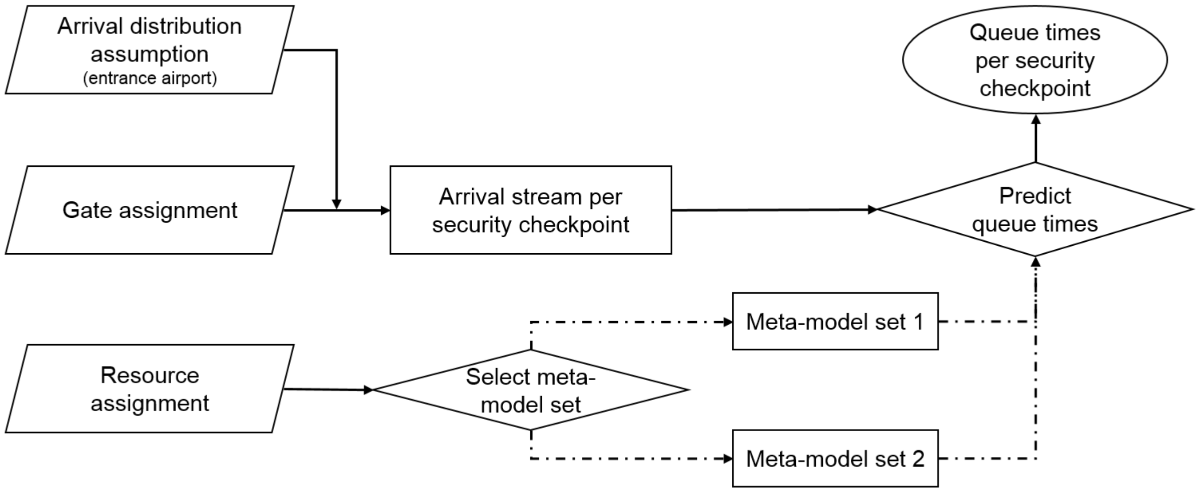


Figure 3.3: Functional flow diagram of the queue time prediction using meta-models.

The construction of the meta-models starts by generating a data set on which the meta-models can be fitted. The method used to generate the data set is similar to the approach taken by Miyoung Shin et al. (2014). Basically, what should be done is visualise all the different possibilities of gate assignments. These can range from really heavy assignments in which two large aircraft are assigned close after one another and on the other hand zero assignments. Then uniformly across the figurative distribution space draw X experiments. The data generated by these X experiments (gate assignments) is then pseudo-randomly split for reasons that will be explained in section 6.3. One half will be used for meta-model fitting and the other half was used to asses the general ability (generalisation error) of the models to predict the queue time. This trade-off between fitting error and generalisation error is called the bias and variance dilemma which will be treated in section 6.4. Low complexity models (models with little variables) tend to have a high bias and low variance, while complex models have low bias but high variance. This trade-off is comparable with the trade-off between fitting and generalisation error. Hence, the generalisation error will be assessed on a separate data set (validation data set) which is different from the data set used for fitting.

The two meta-model types that will be created are polynomial response type of meta-models and radial basis function meta-models. The polynomial response meta-model was taken as the current research is the first (to the authors knowledge) to apply meta-modelling to such a large part of an agent-based model. Therefore, this study chooses a fairly basic model (polynomial response model) and a slightly more complex non-parametric model (radial basis function model). In addition, it was shown in literature that the radial basis function model was able to approximate a simple M/M/1 queueing theory model with good precision.

Once the meta-models have been defined, the performance of these models are tested by means of fitting errors and generalisation errors. Finally, the agent-based model meta-models will be integrated with the flight-to-gate and resource assignment problem.

3.6.4. Validation and verification of the models and methods

Validation and verification in the light of current research can be divided into three parts. 1. Do the meta-models represent the agent-based model, 2. Do the meta-models reflect real queue behaviour, 3. Verify that the integrated models (meta-model based optimisation) optimise for the defined objective.

Validating whether the constructed meta-models are able to represent the agent-based model will be done in two ways (following Snee (1977)). The first is done by simulating a random scenario (gate and resource assignment) in AATOM. The same scenario is used to predict the queue times using the set of meta-models. Finally, the simulation results are compared to the meta-model predictions. The difference between the prediction and simulation realisation will give an indication of the validity of the meta-models.
Secondly, a case study will be performed to assess the validity of the meta-models in the optimisation strategy. The direct integration of the agent-based model into the flight-to-gate optimisation will, if implemented correctly, result in the actual optimum assignment. Therefore, it should be checked whether the meta-model based optimisation will predict the same optimal assignment.

Validation tests to assess the capability of the meta-models to represent realistic queue behaviour are difficult. The current research focuses on a fictitious airport, and therefore there is no data available of the actual airport. Kleijen (1999) argues that validation should be carried out by performing sensitivity analysis, to find out whether the model contradicts qualitative expert knowledge. This can be done by local or global sensitivity analysis. Both methods have advantages and disadvantages. A local sensitivity analysis shows locally what influence a certain parameter has on the output parameter. This is, in models with many combined variables, strongly related to the settings of the other parameters. Therefore, global sensitivity analysis is preferred, but the method proposed by Saltelli et al. is only capable of assessing the importance of the variables (not the direction). Since the models that will be proposed have combinations of variables in them, global sensitivity analysis will be carried out. This type of sensitivity analysis makes the validation with reality difficult.

The research is concluded by performing a verification case study. This case study will only be tested on the meta-model based optimisation. It can be confirmed that the meta-model based optimisations are implemented correctly, if both models find an optimum gate and resource allocation that will minimise the difference in queueing times among the different security checkpoints.

4

Multi-agent model set-up

As explained in chapter 3, the central modelling and simulation environment in this thesis research is AATOM. It has been programmed in such a way that studies can be performed with it and allows for changes to the environment, agents and interactions. As the focus of current research is on an airport like AAS, changes had to be made in AATOM to accommodate the current study purpose. This chapter will address the changes made in AATOM starting with the airport environment. Furthermore, the passenger diversity and autonomous decision making will be elaborated on.

4.1. Fictitious airport environment

In chapter 3 it was observed that departure hall 1 of AAS showed great potential to serve the purpose of current research. Therefore, departure hall 1 of AAS (connected to wings C and D) would be the basis on which a fictitious airport design would be based.

One of the key features of AAS is the fact that the airport is split into two parts: Schengen and non-Schengen. At the Schengen side, flights to a fellow Schengen agreement country depart. Passengers at the Schengen side do not have to pass border control and therefore there is no passport control processing point in place. On the other side is the non-Schengen area. Logically, on this side of the airport passengers embark on flights towards non-Schengen countries. The passport control at AAS is carried out by Royal Netherlands Marechaussee. Note, that under no-circumstances flights from or to non-Schengen countries are allowed to depart from the Schengen side of the fictitious airport.

The fictitious airport created had four gates: Two Schengen and two non-Schengen gates. One of the main assumptions of this thesis is the following:

Assumption 1 (Path restriction) *Passengers heading for a specific gate, take a specific path corresponding to that gate.*

The main reason for this assumption is due the fact that there needs to be a causal relation between processes at the experimental airport when applying parametric models. Imagine that the flight-to-gate assignment was completely unrelated to the passenger walking paths. The paths taken by the individual passengers are determined by the passengers themselves and could be up till a certain level independent of the assigned gate. Then there would be no in advance known or little correlation between gate assignment variables and passenger process outputs. Since the first meta-model that will be built in chapter 6 is from the family of parametric models, it is required that there is a correlation between the gate assignment and the walking paths. Therefore, it is necessary to have the path restriction assumption.

It was decided that for every gate there would be separate passenger processing points in place. Hence, four gates require four security checkpoints, and four check-in facilities. However, since the Schengen part of the airport does not require passport control, only two passport control facilities are in place for the Non-Schengen gates.

The interior dimensions of AAS were confidential and therefore the designed experimental environment in AATOM was based on educational guesses using rough airport maps. The final airport layout is shown in fig. 4.1.

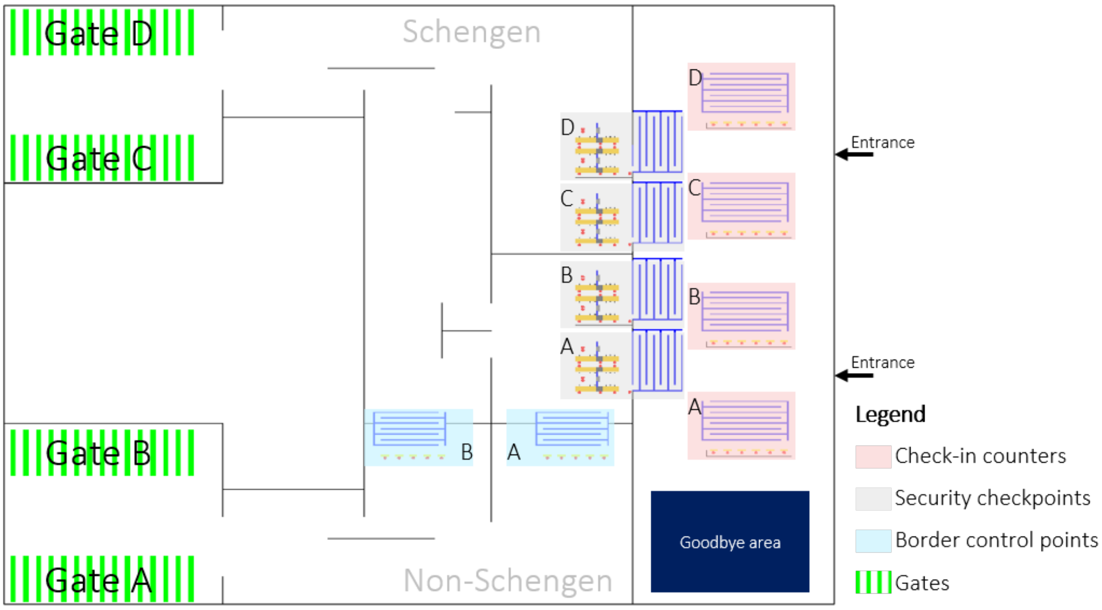


Figure 4.1: Designed fictitious airport in AATOM.

In the figure, the two sides of the airport can be seen. The lower area of the airport is the non-Schengen part and the upper area separated by the border control points is the Schengen area. In the same figure, assumption 1 can be seen. Passengers heading for their flight at gate A will have to pass their designated passenger process points (indicated with *A*).

4.2. Agent specification

In this section the agents present in the ABMS will be discussed. As the model is based on the existing AATOM, only the changes to the existing model will be discussed. For this thesis to be able to draw conclusions about the ability to integrate an agent-based model with an operations research gate assignment optimisation, it is necessary that AATOM would possess all the required properties of an agent-based model. Recall, that for a model to be called an agent-based model three aspects are necessary to be present:

- Interaction among agents
- Local autonomy
- Diversity of agents

These three aspects create the strength of an agent-based model to capture emergent phenomena. The latter two characteristics had to be strengthened as these were not yet clearly present in AATOM.

4.2.1. Passenger diversity

In the baseline AATOM there was a low diversity of passengers present. Each passenger had the same characteristic and had little to no local autonomy. Therefore, it was decided to increase the level of diversity among agents by making a distinction in passenger's purpose of travel.

The two commonly known passenger types are:

- **Business passengers**: Passengers that are comfortable with travelling by aircraft for work. They spend little to no time shopping and mostly do not dwell around the airport.

- **Leisure passengers**: Passengers who travel by aircraft to their holiday destination, friends or relatives.

In a paper by Schultz and Fricke (2011), the managing of passenger handling at airports is studied. This paper zooms in on the fact that people with different purposes of travel have different amount of luggage with them, different walking paces and different arrival distributions. The evidence found in this paper has been implemented in AATOM.

Business and leisure passengers' perceptions directly depend on the individual's system experiences. In addition, the individual's system experience also affects the decisions made by the two types of passengers. Business passengers are familiar with all systems of the airport and usually do not dwell around the airport.

The paper by Schultz and Fricke examined 595 passenger movements at Dresden Airport. One of the examined characteristics of passengers (using video-based passenger tracking) was the walking speed of passengers with different travelling purposes. The empirical results found are shown in table 4.1.

In the table, the different walking speeds for business and leisure passengers are displayed. It was

Table 4.1: Measured speed profiles for different passengers configuration indicated by expected value (μ [m/s]) and standard deviation (σ [m/s]), table has been taken from Schultz and Fricke (2011).

Group	Business		Leisure		Average	
size	μ	σ	μ	σ	μ	σ
1	1.38	0.21	1.19	0.25	1.36	0.23
2	1.17	0.17	0.97	0.20	1.06	0.21
3	1.04	0.23	0.93	0.17	0.96	0.19

observed at Dresden airport, that the business purpose passengers on average walk faster than leisure passengers. Also, except in the case of groups size equal to three, the standard deviation of the walking speed is also lower for business passengers.

In the current state of AATOM group behaviour has not yet been taken into account. Therefore one should pay attention only to the first row in table 4.1. As there is little data available similar to the Dresden airport data, it will be assumed that the results found will also hold for the fictitious airport in current study. Hence, the first diversification of agents is made by establishing business and leisure passengers with mean walking speeds of 1.38 m/s and 1.19 m/s, respectively. Furthermore, the standard deviations of these mean walking speeds were also taken from table 4.1.

Another major difference between business and leisure purpose travellers, is the fact that they arrive at the landside with different distributions. Schultz and Fricke show that leisure passengers arrive from three hours before departure till 30 minutes before departure. The observed patterns from Dresden Airport can be found in fig. 4.2. The arrival of leisure passengers seems to be normally distributed around 1:40 [h:mm] before departure. On the other hand, business passengers tend to arrive at the landside around 1:10 [h:mm] before departure. The distribution for business passengers does not appear to be normally distributed.

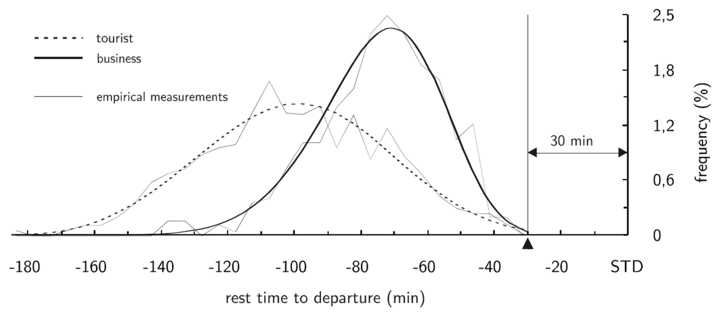


Figure 4.2: Arrival distribution of business and tourist purpose passengers, taken from Schultz and Fricke (2011).

These different arrival distributions needed to be implemented, such that the agent-based model used for integration represents reality as much as possible. Hence, the arrival distributions have been simplified and recreated by approximating the curves in the above graph. The simplified arrival distributions (deduced from fig. 4.2) are shown in fig. A.1 in appendix A. In fig. 4.3, an example is given of the assumed number of arriving business and leisure passengers for a flight of 500 passengers over time. The passengers will arrive following a Poisson distribution.

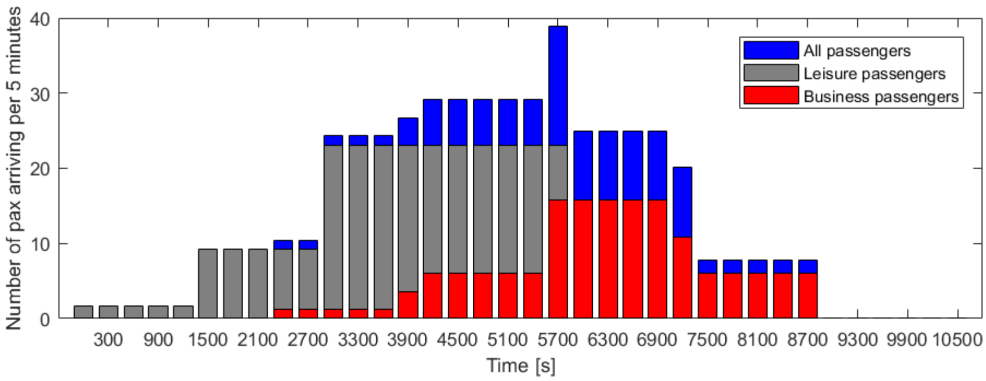


Figure 4.3: The number of passengers arriving following a Poisson distribution per 5 minute interval for a flight that departs after 10800 seconds.

The distributions for landside arrival are however not the only arrival distributions that have to be taken into account. Transferring passengers form a very large group of passengers at major hub airports. At Amsterdam Airport Schiphol, the transferring passengers in 2016 were 37.8% of all departing passengers (Schiphol Group (2016)). Therefore it was assumed that for every departing flight on the non-Schengen side of the airport, 37.8% of the arriving passengers would be transfer passengers. Transfers from non-Schengen flights to Schengen flights have not been taken into account. The reason for this is that at this point (opening the research area) the amount of extra complexity would not add value to the analysis. Since these type of passengers would be the only ones using the border control processing point from the non-Schengen area to the Schengen area.

Concluding, for a flight that departs at the non-Schengen side of the airport 37.8% of the total passengers headed for that flight are transfer passengers. The other 62.2% is spawning at the entrance of the airport. Furthermore, the ratio business/leisure passenger has been taken from Schiphol Group (2016) as well. In here, it was stated that 32% of the total passengers has a business purpose of travel.

The decisions made above lead to the following assumption:

Assumption 2 (Passenger knowledge) *It is assumed to be known in advance, how many passengers will depart per aircraft as well as their arrival distribution.*

Furthermore, passengers have the ability to decide whether they would like to check-in online or pass by one of the check-in desks. Whether passengers need to pass by the check-in desk is also dependent on the luggage type they are bringing. A passenger with only carry-on luggage does not need to pass the check-in desk if he/she is checked-in online. However, if a passenger is checked-in online but is bringing checked luggage, he/she needs to go by the check-in desk. The ratio of online check-in versus desk check-in and carry-on over checked luggage have been set equal for both passenger types. In reality this figure might be different, but this would not change the developed methodology in this thesis.

Another characteristic that has been implemented to strengthen the diversification, is the bag complexity factor. The bag complexity factor gives an complexity indication of the bags that are brought through the security checkpoints. A higher bag complexity level means that the security operators need more time to check it. However, due to the fact that the function has not yet been fully implemented in AATOM, it will not have an effect during the current research.

The entire semi-formal multi-agent simulation specification of AATOM in the LEADSTO language is not given here. An elaborate specification of the AATOM model can be found in the work by Janssen et al. (2017).

4.2.2. Autonomous decision making

The other aspect that needed to be strengthened was the local autonomy in the system. In the initial state of AATOM, people could already take various autonomous decisions. Passengers could decide to check-in online, visit restaurants, or go shopping. The decision was taken to introduce extra local autonomy by giving the agents the ability to go towards a '*Goodbye area*' (see fig. 4.1).

The goodbye area represents a place where passengers could say goodbye to people who dropped them off, or to have something to eat or drink before they would continue towards the security checkpoint. Hence, the agents can decide to go there before going to the designated security checkpoint.

The decision making is based on the paper: "*Time pressure and stress in human judgement and decision making.*" (Maule and Svenson (1993)). The stress is caused by uncertainty and queue build-up and the risk of not making the flight. Whereas, the uncertainty is caused by the fact that the passenger does not have complete information of the time it takes to go through security. The passenger knows its time till departure and makes an estimate about its excess time. The excess time is the time till departure minus the time needed to pass security and border control. In the current study it is assumed that if the time till departure is less than one hour, then there is no time to visit the goodbye area. If one decides to visit the goodbye area, then the maximum time it stays here is drawn from the normal distribution (N(600,200) [s]).

The queue build-up is another factor that influences the decision-making of the passenger. Initially, when the passenger enters the terminal building it observes a certain 'crowdedness' at the airport security checkpoint. Hence, it is assumed that the passenger is able to observe its designated security checkpoint queue length at all times. It could be the case that the queue observed is so long that the threshold is being exceeded at which the passenger believes it needs all its excess time to pass all the necessary checkpoints. Based on many simulations the decision was made to set the threshold to a queue length of 25 passengers.

Furthermore, once the passenger has decided to visit the goodbye area, it checks it with regular intervals to see if the queue' state has changed. The observed queue state can be:

- Worsened: The queue length has increased with respect to the initial observation.
- Neutral: The queue length has remained unchanged with respect to the initial observation.
- Improved: The queue length has decreased with respect to the initial observation.

This regular check takes place with respect to the previous observation after 2.5, 5.0, 7.5 and 10.0 minutes. When the observed state changes to *worsened*, the passenger decides to immediately go to the security checkpoint. This reflects the stress increase caused by queue build-up.

Finally, not every passenger takes the time to visit the goodbye area. Two assumptions have been made on this. Business passengers in general do not spend time before going through the security checkpoint (Schultz and Fricke (2011)). Partly caused by the fact that they enter the airport 'late'. Therefore, it was assumed that the business passengers would not visit the goodbye area. Secondly, not every leisure passenger will visit the goodbye area. Therefore, it was assumed that the probability that a leisure passengers would visit the goodbye area equals 0.3. Formally, the start of the goodbye activity for a passenger (among the 30%) is modelled as follows:

$$
\begin{array}{lrl}
 & \texttt{activity\_area(Goodbye\_activity)} \in & \texttt{obs(current\_area)} \\
\wedge & \texttt{obs(activity\_area\_free)} = & \texttt{true} \\
\wedge & \texttt{obs(queue\_length)} \leq & 25 \\
\wedge & t < & t_{departure} - 1\texttt{hr} \\
\rightarrow & \texttt{activity\_state(Goodbye\_activity)} = & \texttt{in\_progress}
\end{array}
$$

The implemented diversity and local autonomy defines multiple walking routes of the passengers. These walking routes have been graphically displayed in fig. 4.4.

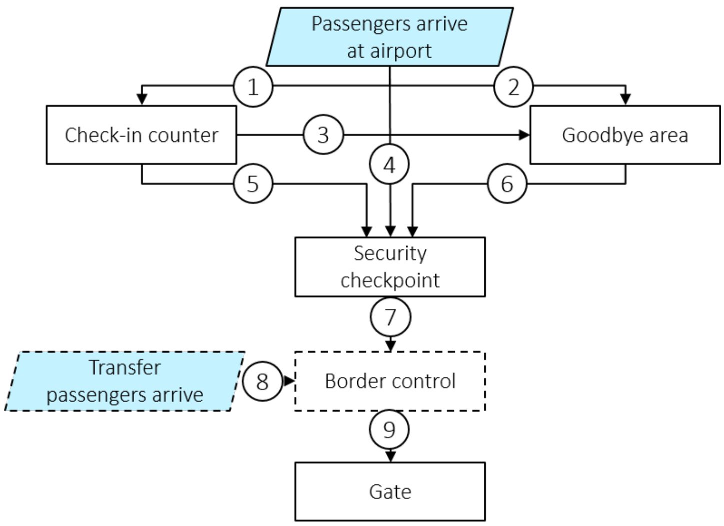


Figure 4.4: Flow chart of possible walking routes to a (non-)Schengen gate.

As can be seen from the figure, passengers arriving at the landside could take take four different paths to the security checkpoint by taking the following arcs:

Route 1 : (4)
Route 2 : (1),(5)
Route 3 : (2),(6)
Route 4 : (1),(3),(6)

Note, that route three and four are never taken by business passengers as they do not go to the goodbye area. Leisure passengers can take all paths depending on their luggage, check-in status, and their observations/decisions.

4.2.3. Sizing the facilities

Finally, before starting on the integration of the agent-based model with the operations research optimisation, the amount of operators for every processing point need to be determined. In chapter 1 and in the work by Schultz and Fricke (2011) it was mentioned that the security checkpoints were the major landside sources of delays. In a study by Eurocontrol (2017) it is found that 5-12% of the delayed flights due to landside terminal elements originate at the security check. Therefore, the number of operators are chosen, such that the bottleneck at the airport would be at the security checkpoint.

In order to determine the amount of operators needed to handle passengers, it is necessary to estimate the amount of passengers that will arrive daily at the airport. The amount of passengers arriving at the airport depend on the number of passengers per flight and the maximum amount of flights that can be scheduled within a certain time frame.

First, the different sizes of aircraft that depart from Schiphol airport every day are analysed. This analysis was done using data of the 9th of October 2017 taken from (https://www.flightradar24.com/). Between 9:00 AM and 10:00 PM 517 flights departed from Schiphol airport. Of these flights, 231 were Schengen flights and 286 were non-Schengen flights. In the table below (in table 4.2), the Schengen and non-Schengen flights are grouped according to their size.

Table 4.2: Departure size of aircraft at Amsterdam Airport Schiphol, 9 October 2017 between 9:00 AM and 10:00 PM.

Flight type	Aircraft size [#seats (% of total)]			
	≤100	(100,200]	(200,300]	>300
Schengen	62 (27%)	167 (72%)	2 (1%)	0 (0%)
Non-Schengen	45 (19%)	132 (57%)	57(23%)	57 (25%)

What can be observed is that, for both Schengen and non-Schengen flights, the majority of the departing aircraft has between 100 and 200 seats. However, 25% of non-Schengen flights have more than 300 seats.

Secondly, the maximum number of departures in the planning horizon needs to be determined. Together with the aircraft sizes, the maximum number of passengers per time interval can be determined. This information is needed for determining the minimum required number of operators per passenger processing point.

First, the interval standard needed to be set for the entire project.

Assumption 3 (Time interval standard) *The time interval over which passenger arrival rates and queueing times will be given is set to 5 minutes.*

This interval has been set such that enough people will arrive per interval and enough data can be gathered to give a good indication of what happens within the interval and between intervals. In fig. 4.5 the queueing time is displayed for all passengers. In this figure it can be seen that the queueing time does almost not change within a five minute interval, but slightly changes between intervals. Hence, the five minute interval is taken as the time over which the queue times will be averaged.

In order to determine the maximum amount of arriving passengers per time interval, it is necessary to know at what frequency aircraft depart. Hence, the smallest turn around times with the largest aircraft could lead to the highest amount of passengers entering the airport during a specific interval. It was found in literature and on internet resources (https://www.flightradar24.com/) that the quickesst turn around time for aircraft in group >200 (table 4.2) was little over 1 hr. Furthermore, the quickest turn around times found for aircraft of size ≤200 was around 20 minutes. These two extremes were used to find the maximum amount of arriving passengers per time interval.

The first extreme were two departing aircraft with 500 passengers aboard (A380), of which the second aircraft departs 1 hour after the other has departed. This scenario will never occur in real live but it was useful to examine the queue times of this scenarios. This first extreme scenario a maximum of around 49 passengers arriving in five minutes would be observed. This was determined using the arrival distributions created as was shown in fig. 4.3. The second extreme consist of two aircraft with 200 passengers aboard, of which the second aircraft departs 20 minutes after the first aircraft has departed. The maximum arriving passengers within five minutes was a little over 48 passengers (in five minutes).

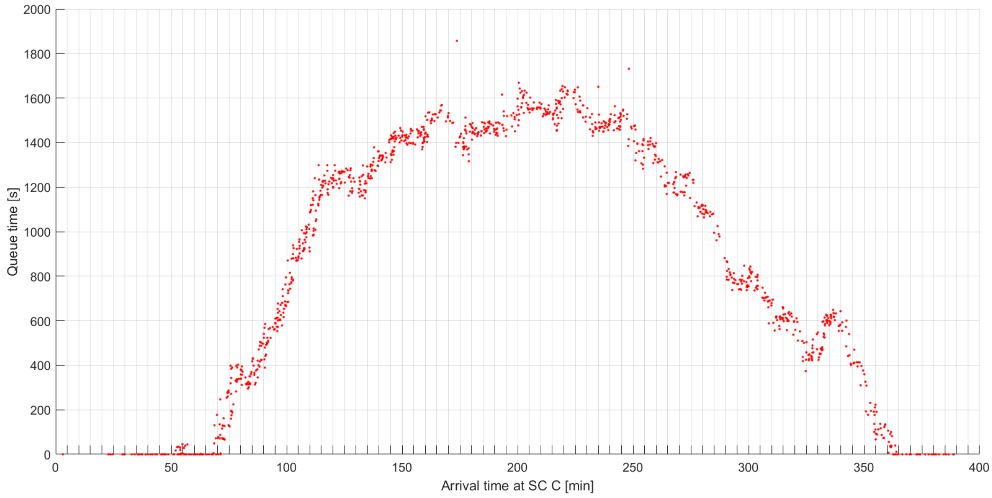


Figure 4.5: Arrival time of passengers at security checkpoint C versus the queue time experienced by the passengers.

These two scenarios, together with other scenarios were simulated in AATOM. Main conclusions about the observations was that the queue build up was extremely sensitive to the number of X-ray scanners used. The extreme scenarios resulted in peaks where major queue build up occurred, when two X-ray scanners were opened. However, when three X-ray scanners were used, the queue build up almost completely disappeared (which will be seen in fig. 5.2). When this was discovered it was decided to include the possibility to keep the number of X-ray scanners open. Hence, this would be one of the decision variables in the problem statement.

The other facilities have been designed in a similar fashion, except for the fact that the objective was to avoid bottlenecks at these facilities. Hence, the check-in facilities and border control facilities have been slightly over-sized.

To conclude this chapter, an overview of the parameters implemented in AATOM are listed in table A.1 in appendix A. Note that not all these settings have been calibrated yet with real airport data. Therefore, these settings shall only be used to replicate the current study or could be used for future research.

5 Simulation optimisation

The initial way to integrate the agent-based model and simulation with an operations research gate assignment optimisation would be to connect both paradigms in a direct manner. In this chapter the initial integration is made using a simulation optimisation method, which was explained in chapter 2.

The method proposed has been deduced from the paper by Casado et al. (2004). In the paper a decision support system for optimising passenger flow is made. This system makes the trade-off between service quality and labour costs at an airport. The paper integrates a simulation module with an optimisation module that requires that Dantzig's labour scheduling problem is solved. The difference between the reference paper and the current study is the fact that in the reference paper the problem needed to be solved in short computational time. Ultimately this would be the case for the current study as well, but for an initial study to open the scientific gap, this seems one step too far. Hence, this paper has a similar aim as the current research and therefore the steps made in this paper will be applied to the current research.

The first step will be to define the problem that needs to be solved in section 5.1. In section 5.2, the optimisation algorithm will be explained. The chapter is concluded by reflecting on the created simulation optimisation method. The implementation of the integration method will be elaborated on, and in chapter 7 a small case will be solved.

5.1. Non linear programming definition

In this section the problem will be defined that will be solved by simulation optimisation. First, the general non linear programming definition will be given and during the section the problem will be customised to fit the current problem at hand.

Let n, m, and p be positive integers. Let X be a subset of R^n, let f, g_i, and h_j be real-valued functions on X for each i in 1, …, m and each j in 1, …, p, with at least one of f, g_i, and h_j being non linear.

The general form of a non linear programming problem is displayed below.

$$
\begin{aligned}
\underset{x}{\text{minimise}} \quad & f(x) \\
\text{subject to} \quad & g_i(x) \leq 0, \; i = 1, \dots, m. \\
& h_j(x) = 0, \; j = 1, \dots, p. \\
& x \in X
\end{aligned}
$$

The problem at hand in the current study is a flight-to-gate assignment. In the problem definition above the function f will, in the simulation optimisation method, be an expectation of the simulation.

Mangoubi and Mathaisel (1985) wrote a simple gate assignment model which will be the basis for current study. The difference between the current gate assignment problem will be the objective function.

Whereas Mangoubi and Mathaisel were optimising for passenger walking distances, the current research will focus on optimal utilisation of the security checkpoints. This means that the objective is to evenly distribute waiting times per time interval among the security checkpoints.

The parameters that are of importance:

- N is the complete set of flights considered,
- V is the set of Schengen flights, which is a subset of N,
- M is the complete set of gates considered,
- U is the set of Schengen gates, which is a subset of M,
- n is the total number of flights,
- m is the total number of gates,
- s is the total number of X-ray scanners open during the planning horizon.

The first (binary) decision variable x_{ij} is assigned for each possible flight-to-gate assignment, where:

$$x_{ij} = \begin{cases} 1 & \text{if flight i is assigned to gate j,} \\ 0 & \text{otherwise.} \end{cases}$$

The second decision variable is the integer r_j and represents the number of X-ray scanners open at security checkpoint j, which is the security checkpoint of gate j.

However, it was decided in section 4.2.3 to constrain r_j to only two possible values.

$$r_j = \begin{cases} 2 & \text{if 2 X-ray scanners are open at security checkpoint j,} \\ 3 & \text{if 3 X-ray scanners are open at security checkpoint j.} \end{cases}$$

In eq. (5.1) the model of Mangoubi and Mathaisel (1985) is applied to the current problem.

$$\begin{aligned} \underset{x_{ij},r_j}{\text{minimise}} \quad & f(x_{ij}, r_j) = \sum_{int=1}^{T} \frac{1}{m-1} \sum_{j=1}^{m} (SCQT_{int}^{j}(x_{ij}, r_j) - \overline{SCQT_{int}}(x_{ij}, r_j))^2 \\ \text{subject to} \quad & \sum_{j \in U} x_{ij} = 1, && \forall\, i \in V \\ & \sum_{j \notin U} x_{ij} = 1, && \forall\, i \notin V \\ & \sum_{h \in L(i)} x_{hj} + x_{ij} \leq 1, && \forall\, i \in V \quad \forall\, j \in U \\ & \sum_{h \in L(i)} x_{hj} + x_{ij} \leq 1, && \forall\, i \notin V \quad \forall\, j \notin U \\ & \sum_{j \in M} r_j = s \end{aligned} \tag{5.1}$$

In the above problem the objective function is to minimise f, which is a function of x_{ij} and r_j. The objective of the current minimisation problem is to make sure that the utilisation of the security checkpoint queues is spread among the facilities. In reality this would mean that the objective is to make sure that the waiting times are spread equally among the stations. The thought behind this is that once the queueing times are equally spread among the checkpoints, then the overall passenger experience at the checkpoints is optimised. Furthermore, delays caused by congestions at the security checkpoints are minimised.

Currently, passengers headed for different gates pass through the same set of security checkpoint facilities. In reality, queue build-up is managed ad hoc. This is done by an assigned security officer that assigns groups of persons to a specific security checkpoint or X-ray machine. The idea of current research is to direct the passenger streams in a smart way, such that the congestions are kept to a minimum. By simulating the passenger streams in a optimisation loop, different scenarios can be evaluated. Finally, the optimisation algorithm will explore promising scenarios and eventually select the 'optimal' solution. This method is not suitable to be solved by exact solution algorithms. Hence, the brackets around optimal, since these methods could converge to local optima.

The objective function will be to minimise the sum of variances of the security checkpoint queue times during the time intervals, as shown in eq. (5.1). This objective function will make sure that the queue times experienced by passengers at the different security checkpoints are as close as possible to each other during every time interval int. The optimal assignment of flights-to-gates and resources will make sure that the passenger experience at the security checkpoint is optimised. An illustrative example of the four observed security queue times during five minute intervals is shown in fig. 5.1. The security checkpoint queue time ($SCQT$) is defined as the average time that passengers are waiting in a queue before they can proceed to an X-ray scanner during the five minute interval. The variance of security checkpoint queue times is determined per time interval and then summed to arrive at the objective function value.

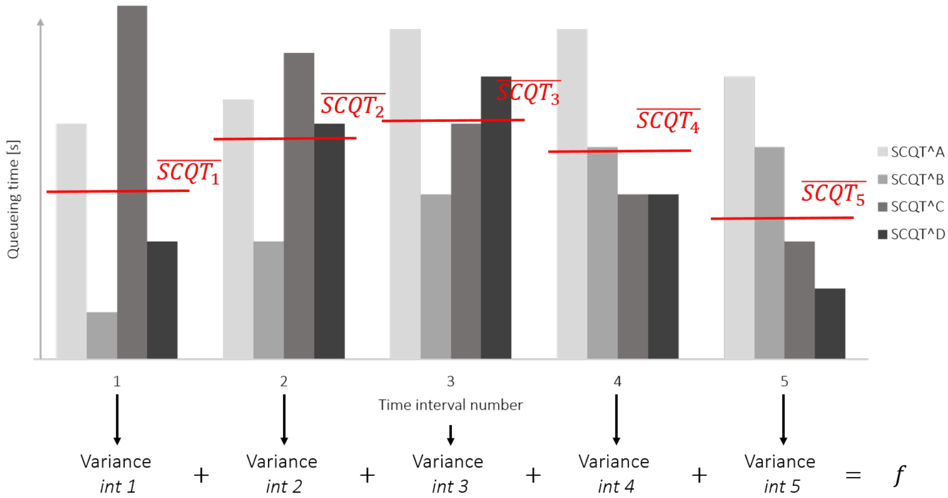


Figure 5.1: Illustration of the objective function calculation.

The calculated variance of a single interval is a measure for the spread of the queueing times among the different checkpoints during that interval. Summing all the calculated variances will give a measure for the overall spread. The variance per interval will be calculated by:

$$var_{int} = \frac{1}{m-1}\sum_{j=1}^{m}(SCQT_{int}^{j} - \overline{SCQT_{int}})^2, \tag{5.2}$$

where $\overline{SCQT_{int}}$ represents the average of the security queue times of the m security checkpoints during the time interval int. Then the objective value f is calculated using eq. (5.3):

$$f = \sum_{int=1}^{T} var_{int}, \tag{5.3}$$

where T represents the total number of time intervals in the examined planning horizon.

In the simulation optimisation method, the security checkpoint queueing times will however not be calculated by means of equations. These times will be the outcome of many simulations. Due to the stochasticity in the model (pseudo random number generators), Monte Carlo simulations need to be performed (at least ten times) to be able to say something about the queue behaviour during one of the intervals. The number of replication runs was set to ten, since after ten runs the variance of the queue times per time interval did not change much any more. Hence, running more simulations of the same scenario did not give more information about a scenario. After these Monte Carlo simulations, the average queueing times per interval per checkpoint are calculated and fed back to the optimisation algorithm. The algorithm can try to reduce the objective function value f by changing the flight assignment (x_{ij}) or by assigning extra X-ray scanners (r_j).

The problem is constrained by four types of constraints of which two were used by Mangoubi and Mathaisel (1985). These two have been split-up since the current problem handles two types of flights (Schengen and non-Schengen). Schengen flights cannot be assigned to non-Schengen gates and vice versa. When formulating the constraints in eq. (5.1) this has been taken into account by splitting the assignment problem constraints in two sets.

The initial set of two constraints in eq. (5.1), makes sure that every flight is assigned to one and only one gate. The number of constraints of this type are equal to the number of flights considered n. The second set of constraints makes sure that no two aircraft may be assigned to the same gate concurrently. In the first constraint of the set, $L(i)$ represents the set of all Schengen flights h which landed before flight i and are still on the ground at the time flight i arrives. Hence, $L(i)$ represents the Schengen conflict set and if the flights are indexed in order of their arrival time it can be defined as:

$$L(i) = \{h \mid t_h^d \leq t_i^a,\ h = 1, \dots, i-1\ \forall\ i \in S\} \tag{5.4}$$

where t_h^d is the departure time of flight h and t_i^a is the arrival time of flight i. Since it has been mentioned that only departure flights are considered, an assumption needs to be made about a time window for a flight to be present at the gate. This is the time that a gate is locked for its use by a specific flight. This includes the turn around time of flight i as well as a buffer. This assumption is based on a similar assumption made by Yan and Huo (2001).

Assumption 4 (Turn around time) *The turn around time (including buffer) of an aircraft is dependent on its size (amount of seats). It is assumed that:*

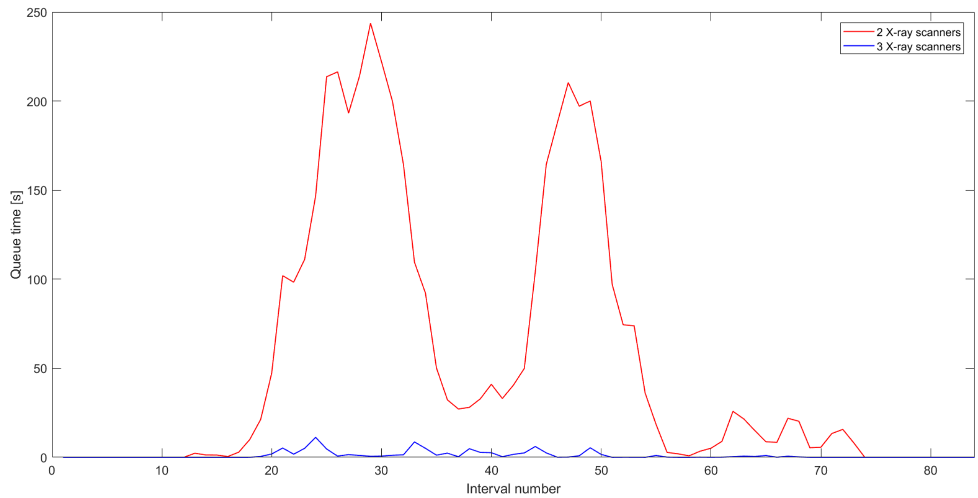
$$Turn\ around\ time(i) = \begin{cases} 2400s & \textit{if flight } i \textit{ has} \leq 200 \textit{ seats} \\ 4800s & \textit{otherwise.} \end{cases}$$

These turn around times have been based on actual turn around times found (https://www.flightradar24.com/), turn around times assumed by reference papers like Diepen et al. (2012), Yan and Huo (2001). Furthermore, the minimal assumed turn around times showed the most interesting queue time behaviours when using two or three X-ray scanners.

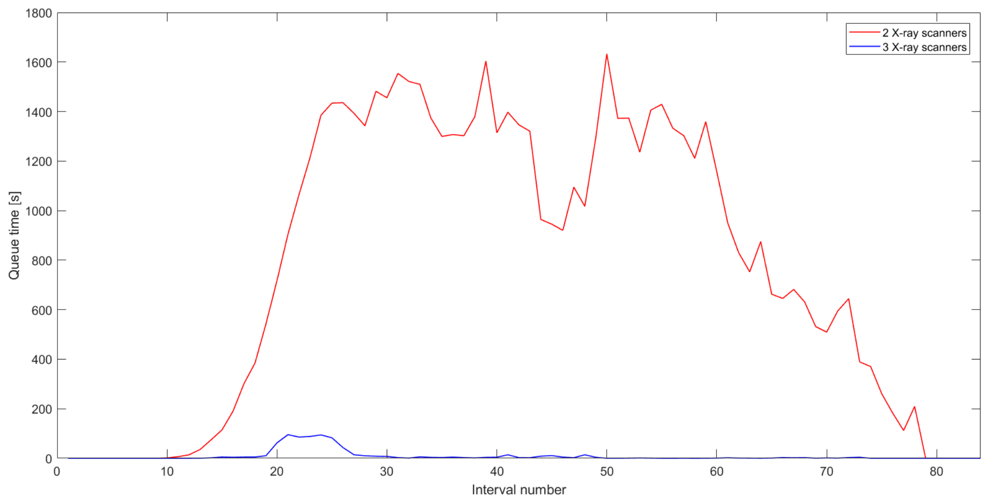
To determine the conflict set, the t_i^a is set equal to the time that an aircraft has the right to occupy a gate and is not necessarily the arrival time of flight i. The similar principle holds for the departure time. In this way a certain robustness is added to the model. However, this does not make the model completely robust to severe changes in the schedule. Though, it can be reasoned that passengers coming in for a flight that is last-minute delayed have already passed the security queue and therefore do not affect the validity of the proposed method.

The final constraint is the resource constraint. Every X-ray scanner requires four operators and it is assumed that that the security agency plans it personnel on its own in advance. Therefore, there is a limited amount of resources that can and need to be used during the day. Hence, this is a pre-assumed number of X-ray scanner s that need to be opened during the planning horizon, as this corresponds to the number of security operators working at the particular day. Since the values of r_j have been restricted to either two or three, the number of total X-ray scanners s is also restricted to a finite set. The reason for implementing this constraint has to do with the fact that currently AATOM is still under development. This means that the personal characteristics of the agents (e.g. individual efficiency) are not yet fully implemented and therefore the queue time behaviour of security checkpoints are fairly similar. What is expected is that the security checkpoint queues of the non-Schengen and Schengen gates will almost behave identically (the data will show little differences). For example, if two identical non-Schengen flights are expected at the same time, the queueing times experienced during the time intervals preceding the arrivals will show little difference between security checkpoint A and B. Setting the $s = 10$ could ensure that there will be a clear asymmetry between the security checkpoints, two security checkpoints could have three X-ray scanners active and the other two could have two X-ray scanners active.

Furthermore, the speed and efficiency of the optimisation algorithm depends on the shape of the objective function. The addition of the resource constraint will make objective function less flat, probably leading to a better performing optimisation algorithm. The impact of different amount of X-ray scanners at security checkpoint C (SC C) is shown in fig. 5.2.



(a) A case with relatively low observed queue times when two X-ray scanners active at SC C.



(b) An extreme case with high observed queue times when two X-ray scanners active at SC C.

Figure 5.2: The impact of using an extra X-ray scanner shown in two cases for security checkpoint C.

5.2. Differential evolution algorithm

The algorithm that will be used to solve the non linear problem is the differential evolution algorithm. The basic idea of the differential evolution (DE) algorithm is similar to evolutionary steps in nature. It has been demonstrated that the DE algorithm converges faster and with more certainty than other global optimisation methods such as a genetic algorithm (Storn and Price (1997)). The initial idea was to use a genetic algorithm implemented in the MATLAB environment. However, the MATLAB implementation of a genetic algorithm fails to cope with the combination of integer programming with equality constraints. The DE algorithm employed in current study, a version of the 'Multiobjective Evolutionary Algorithm Based on Decomposition' (Zhang and Li (2007)) supplied by V. Ho-Huu, could readily cope with the formulated problem.

The method is a parallel direct search method that uses NP D-dimensional vectors

$$x_{i,G}\ ,\ i = 1, \dots, NP \tag{5.5}$$

as a population for every generation G. The algorithm is initialised by randomly choosing the initial population that covers the entire parameter space. Random decisions are made using a uniform distribution.

DE generates new populations by adding the weighted difference between two population vectors to a third vector. This process is called mutation.

Let $x_{i,G}$ $i = 1, \dots, NP$ be a target vector, yields the lowest objective function value. Then a mutant vector is generated using:

$$v_{i,G+1} = x_{r_1,G} + F \times (x_{r_2,G} - x_{r_3,G}) \tag{5.6}$$

with random indexes $r_1, r_2, r_3 \in 1, \dots, NP$, integer, mutually different and $F > 0$. The random integers should be chosen such that they are different from the running index i. Therefore, NP should be at least equal to four to allow for this condition. The real parameter F controls the amplification of the differential variation $(x_{r_2,G} - x_{r_3,G})$, and $F \in (0, 1)$.

The next step is crossover. Crossover increases the diversity of the perturbed parameters. The output of this step will be the trial solution that will challenge the target vector. The trial vector will be of the following form:

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \dots, u_{Di,G+1}) \tag{5.7}$$

In fig. 5.3, the crossover process is illustrated.

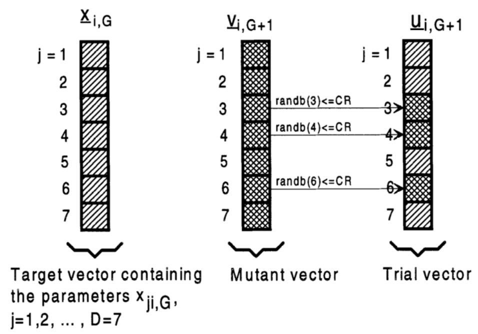


Figure 5.3: Illustration of crossover process, taken from Storn and Price (1997).

As can be seen in fig. 5.3 the trial vector is formed by

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } (randb(j) \leq CR) \text{ OR } j = rnbr(i), \\ x_{ji,G} & \text{if } (randb(j) > CR) \text{ AND } j \neq rnbr(i) \end{cases} \qquad j = 1, 2, \dots, D.$$

where $randb(j)$ is the jth evaluation of a uniform random number generator with an outcome $\in [0, 1]$. Furthermore, CR is the crossover constant and is to be specified by the user ($CR \in [0, 1]$). The function $rnbr(i)$ randomly chooses an index $\in 1, 2, \dots, D$ such that it ensures that at least one parameter of $v_{i,G+1}$ passes on to $u_{i,G+1}$.

Finally, to decide whether or not it should become a member of generation $G+1$, the trial vector $u_{i,G+1}$ is compared to the target vector $x_{i,G}$, using a greedy criterion. If the trial vector yields a smaller objective function value than the current target vector, then $x_{i,G+1}$ is set to $u_{i,G+1}$. If not, then the old target vector is retained. Once the method converges, the iterations are stopped and the optimum is found.

The direct integration of the agent-based model into the differential evolution algorithm created in the current study is shown in fig. 5.4, which is similar to the procedure used by Casado et al. (2004) and Carson and Maria (1997) presented in section 2.3. It includes the steps described above and additional steps to evaluate the trial and target vector. To obtain the objective function values, the target and trial gate and resource assignments need to be simulated in AATOM. This is done by performing Monte Carlo simulations on the NP individuals in the generation and the trial vectors. Monte Carlo simulations had to be repeated ten times per individual in order to be able to say something about the response of the system. Many outputs can be taken from the simulations, but the output of interest for the current study is the queueing time experienced at the security checkpoint per time interval, $SCQT_i$.

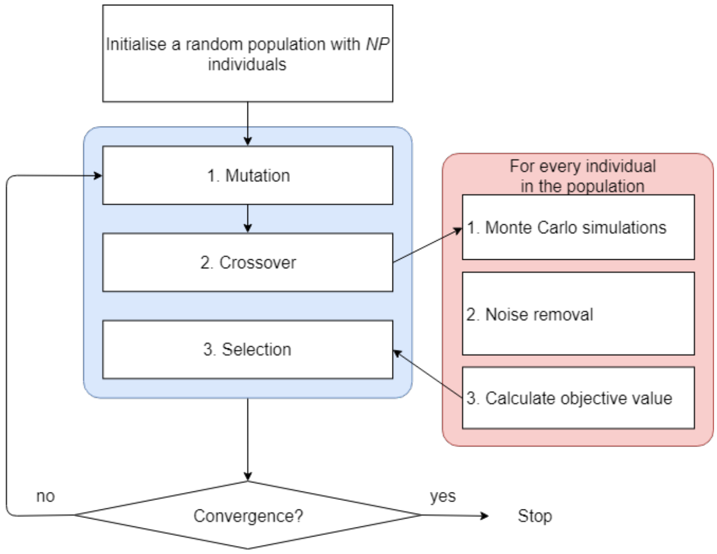


Figure 5.4: Illustration of the integration of the agent-based model into the differential evolution algorithm. (Blue) Differential evolution algorithm. (Red) Simulation and data handling steps.

The above described step requires computational power and time: A 'seven hour day' at the airport, could take around two hours to simulate. Therefore, if the entire generation was of size four (NP = 4, the bare minimum), it would take around 20 hours to evaluate a single individual ten times. Thus, evaluating the entire generation would cost around 60 hours, without taking any delays due to manual work into account. This is a major disadvantage of the direct coupling of an agent-based model with an optimisation.

The complete simulation optimisation routine is shown in fig. 5.5.

The DE algorithm develops, based on the flight schedule, potential solutions (gate and resource allocations). These potential solutions are then simulated in AATOM. Before the objective function value can be calculated from the simulation data, the data needs to be handled such that there are no outliers present in the data and the security checkpoint queue times per time interval are obtained. In the next sub-section the data handling steps, taken to convert the raw passenger data from AATOM into the security checkpoint queue times per time interval, are briefly discussed.

5.2.1. Data handling steps

The AATOM output data from the security checkpoint is not directly usable since it is on single passenger level (as seen in fig. 5.5). An example format of raw ABMS output data is shown in table 5.1.

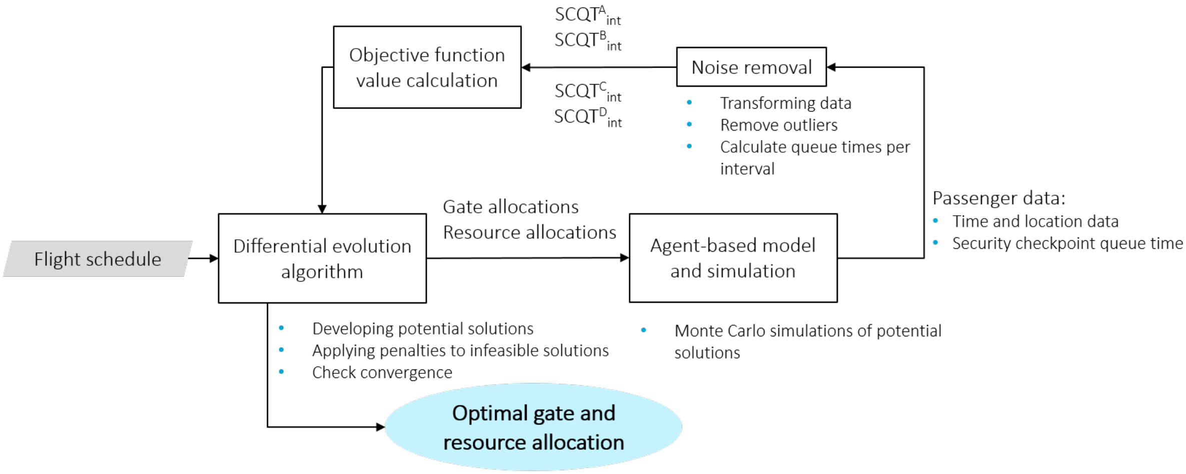


Figure 5.5: Functional flow of the simulation optimisation methodology.

Table 5.1: Example simulation output data.

Time [ms]	Agent ID	Activity
88550	170354826	10
141000	81823678	10
282550	81823678	13
343550	1172147589	10
371550	14615546	14
935200	1947197068	20
1013400	170354826	14

The first column is the time with respect to the start of the simulation and the second column displays the agent identification number. The value in the third column indicates two aspects. The first number, of the two-digit value, is an indicator for the type of passenger. If the first digit is a '1' then the agent is a leisure passenger, while a '2' indicates a business passenger. The second digit indicates the location of the passenger. A '0' is the airport' entrance and '1', '2', '3' and '4' indicate the security checkpoint queues 'A', 'B', 'C' and 'D', respectively. For example, agent ID 170354826 enters the airport at 1min 29s and arrives at security checkpoint D at 16min 53s. Note, that the time is the only indicator that indicates whether the passenger is joining the queue or leaving ($t_{join} \leq t_{leave}$). Hence, the data needs to be managed in such a way that each row contains the data of one passenger (entrance time, security system ID, arrival time, security checkpoint queue time).

After the data set transformation of a potential solution simulation run, the second step is to make sure that the data is free of outliers. *"An outlier is a data point that, because of its extreme value compared to the rest of the dataset, might incorrectly influence an analysis "*(Zuur et al. (2007)). The noise apparent in the simulation data could be a result of agents who are stuck in the environment. What could for example happen is that in busy scenarios certain passengers are pushed into a corner from which they are not able to escape. What is then observed in the data is a very large queue time of that passenger with respect to the queue times experienced by the other passengers. Furthermore, such a stuck agent might influence an other agent's behaviour. Therefore, the raw data needs to be checked and if existent, noise should be removed. The interquartile range (IQR) method is used to remove the noise (outliers) from the data are proposed in the Ecological Data Analysis book by Zuur et al. (2007). In addition to the IQR method, analysing the time between entrance and security checkpoint is another descent way to see if for example an agent got stuck in the environment.

The IQR method identifies outliers by creating box-and-whisker plots of the queue time experienced for every time interval. The edges of the box represent the 1st and 3rd quartile (quartiles are the points that divide the data into four groups of equal size), which are equal to the 25th and 75th percentiles. The interquartile range is the range between the first and third quartile. The IQR method considers a data point to be an outlier if it falls outside either 1.5 times the IQR below the first, or 1.5 times the IQR above the third quartile. In fig. 5.6, box plots per time interval are shown for every security checkpoint of data obtained by ten runs of the arbitrary schedule presented in table 5.2. The lower end of the blue box indicates the first quartile and the top end of the box indicates the third quartile. The red line in the blue box indicates the median of the security checkpoint queue time during that time interval. The whiskers are drawn till the last data point that falls within the 1.5 times IQR range described above. The red markers ('+') indicate the data points (outliers) that do not fall within the range.

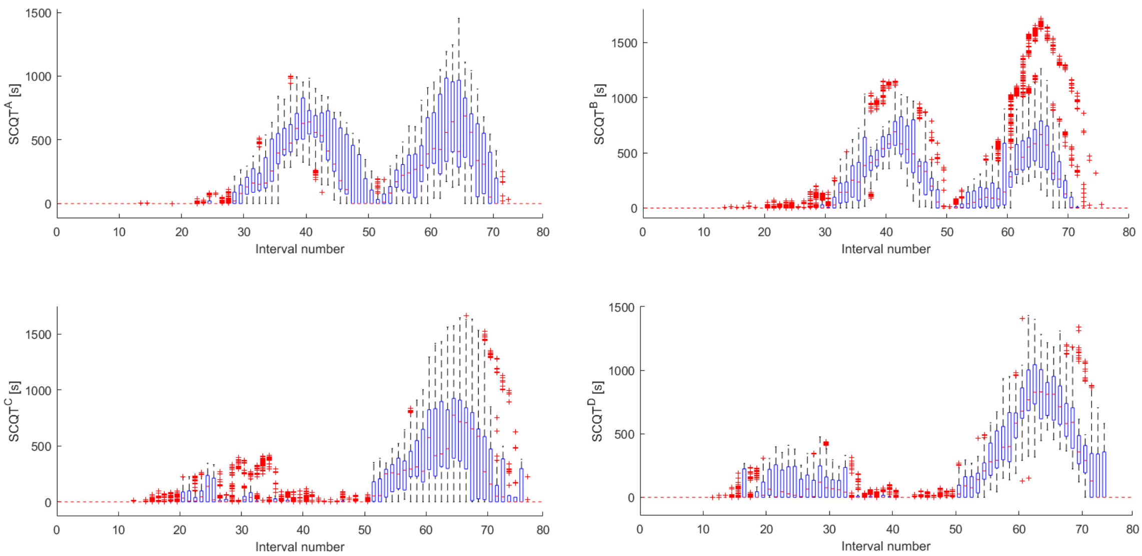


Figure 5.6: Box plots of simulation results for the security checkpoints (l.r.t.b. security checkpoints A, B, C & D), with each two X-ray scanners active.

Table 5.2: Departure times and aircraft sizes used for producing the data of fig. 5.6.

Gate A		**Gate B**		**Gate C**		**Gate D**	
Pax	Time	Pax	Time	Pax	Time	Pax	Time
200	3:00	200	3:00	200	3:00	200	3:00
500	4:20	500	4:20	80	3:40	80	3:40
200	5:00	200	5:00	200	4:20	200	4:20
500	6:20	500	6:20	150	5:40	150	5:40
200	7:00	200	7:00	200	6:20	200	6:20
				200	7:00	200	7:00

As can be seen from table 5.2, the departing flights from Gate A and B are the same as well as the flights from Gate C and D. The time indicated in the table represents the departure time with respect to the starting time of the simulation. Due to the fact that similar schedules were used for the Schengen gates and the non-Schengen gates, the upper and lower two sub-plots in fig. 5.6 show similar behaviour. The box plots show quite some points that fall outside the whiskers per time interval. However, these data points are not necessarily outliers.

Let us go into detail a little bit more by looking at the box plot of security checkpoint B as example. Two different 'outliers' can be observed in this plot.

The first type are the outliers detected where the median of the box is nearly zero. These can be observed before interval number 32. For these time intervals, the median and the 25th and 75th quantile of the box are close to zero (box ends). As a result, the whiskers will be very short. Therefore, if there is a realisation (e.g. one out of the ten runs) that shows higher queue times during these intervals, then the box plot observes those as being outliers. However, these data points are no true outliers, but reflect the stochasticity of the agent-based model and are a result of the pseudo random number generators.

The second type of outlier can be observed in the interval numbers beyond 32 at security checkpoint B. Essentially, the same happens as in the first type of outlier. A small portion of the realisation could show a significantly different outcome due to the stochastic behaviour of the model. This is clearly also the case in fig. 5.6 as the outliers observed form together a different realisation.

Taking into account these two types of misclassified outliers, the data was made noise free. This step mostly included cleaning the data from observations of too high security checkpoint queue times with respect to the rest of the simulation data. An example of a box plot that showed significant outliers is shown in fig. B.1. Finally, the average security checkpoint queue times per time interval could be calculated from the noise free data which is the output of the noise removal step in fig. 5.5.

In chapter 7, the results of the initial direct integration of an agent-based model with an operations research optimisation will be shown. As was mentioned earlier, the simulation optimisation method is a very time consuming method. The size of the problem, number of replication simulation runs and the manual connection between the JAVA AATOM model and the MATLAB optimisation algorithm make it a very slow method. In addition, the specified optimisation strategy would result in a optimal resource and gate assignment for a specific day of operations. If this takes days to generate an optimal assignment then it is impractical to use as a gate planner support tool. This is a well known disadvantage of simulation optimisation and hence the current research into meta-modelling. Therefore, the next step is to see whether the integration of the agent-based model and gate assignment optimisation can be done by means of meta-modelling.

6 Integration by meta-modelling

The optimisation of the direct integration of an agent-based model with flight-to-gate assignment optimisation is a very time-consuming process. Therefore, this chapter will describe a method to integrate the agent-based model with a gate and resource assignment optimisation by making use of an abstraction/meta-model. As explained in section 3.6 this will be done by two types of meta-models: polynomial response functions and Gaussian radial basis functions.

First the input parameters for the meta-models will have to be selected in section 6.1. Then, in section 6.2 the way the data is generated is elaborated on. Finally, in sections 6.3 and 6.4 the meta-modelling steps will be explained for the polynomial response and the Gaussian radial basis function methodology. In these sections the model fitting will be discussed and the performance of the models evaluated.

6.1. Parameter selection

In the early stage (literature study) it was decided to apply meta-models for simulation input-output relations. The first step in creating a meta-model is to determine the parameters that will be used to create the meta-models. In this section, the parameter selection will be explained.

There are numerous variables that can be analysed from the AMBS, but it was decided at an early stage that the security checkpoint queue time ($SCQT_i$) was the observed output in the simulation optimisation method (in chapter 5). Therefore, it was decided to also use the $SCQT_i$ per checkpoint as the output in the simulation input-output meta-model.

As the meta-model will be used to integrate the agent-based model with a flight-to-gate assignment model, it is necessary that these input variables are both related to the flight-to-gate assignment decision variables as well as the simulation output variable (SCQT). Known from queueing theory is that the average queueing time passengers experience is dependent on the arrival rate (at the security checkpoint) and the service rate. However, the arrival rate at the entrance of the security checkpoint queue is not known, due to the fact that there are intelligent agents present at the airport (capable to make autonomous decisions). What is known are the arrival rates of the passenger at the entrance of the airport (shown in chapter 4). Fortunately, these arrival rates of passengers are both related to the flight assignment and the security checkpoint queue time. The arrival distribution can be converted into expected number of passengers entering the airport per time interval using the passenger knowledge assumption. These arrival rates (AR) per time interval i (AR_i) could be used as input variables for the meta-models. Below the variable connections with both paradigms will be further explained.

Relation gate assignment variables with input parameters

The flight-to-gate assignment assigns an incoming passenger stream to a path (recall assumption 1). This conveniently ensures a relationship between the first type of decision variables (x_{ij}) in the flight-to-gate assignment model and the number of passengers per time interval (meta-model input parameters). Hence, this connection is strongly dependent on assumption 1.

Relation input parameters with output parameter

The relationship between the input and output parameters of the simulation is the relationship that will be represented by the meta-model. Hence, it is important that there exists a clear relationship between the input parameter and the output parameter of interest since the initial meta-model that will be developed is a parametric model. One of the agent-based model parameters that could be changed was the amount of X-ray scanners open per security checkpoint (the second type of decision variable r_j). This decision variable directly affects the speed at which a security checkpoint operates and therefore the queue build-up.

Secondly, the main difference between two unique scenarios are the arrival streams of passengers. If one would run several different scenarios (departure schedules), one would observe different queue time behaviour in each of the scenarios. This is as expected, since this behaviour reflects the common understanding of queueing theory.

An example of an arrival stream of passengers (arriving at the airport entrance) and the resulting security checkpoint queue time behaviour is shown in fig. 6.1.

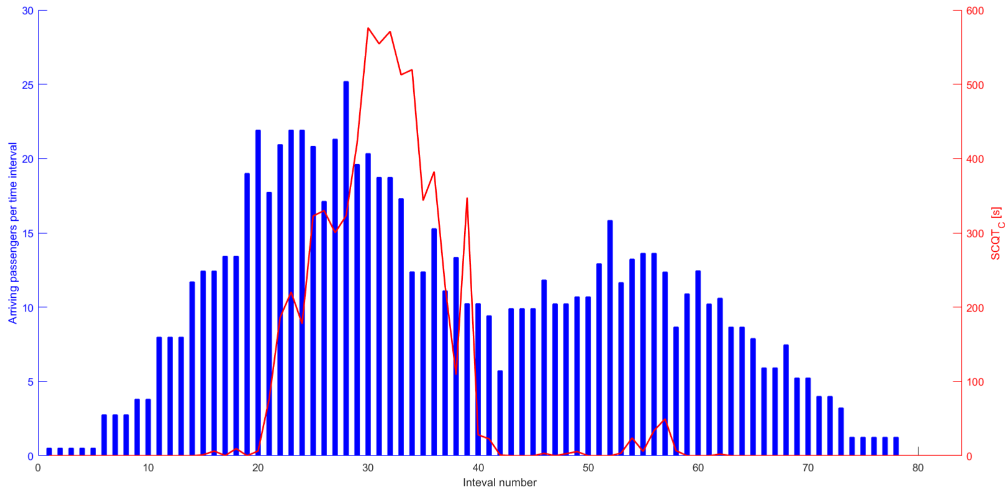


Figure 6.1: Scenario for security checkpoint queue C. (Blue) The number of passengers arriving (at the airport entrance) per interval, headed for security checkpoint C. (Red) The security checkpoint queue times observed per time interval.

The arrival stream as shown in fig. 6.1 is at the airport entrance and this is not equal to the arrival stream at the entrance of the security checkpoint queue. Passenger are intelligent and take different paths to the security checkpoint. They could for example decide to visit the goodbye area, and leave whenever they observe a worsened queue state. Therefore, the clear arrival distribution of passengers at the airport entrance cannot be linearly translated into the arrival distribution at the entrance of the security checkpoint queue. Hence, the well known queueing theory models cannot be confidently used to approximate the security checkpoint queue time, since these models require clear arrival distributions. However, the general queue behaviour (known from queueing theory) is preserved.

So instead of trying to estimate the queue time based on the arrival rate at the security checkpoint queue, the current study will try to find te relationship between the arrival rate at the airport entrance and the security checkpoint queue time. The mathematical form of the relationship is not yet known.

When looking at fig. 6.1, one can see that there is a delay in the effect of the arriving passengers at the airport entrance on the queue build-up at the security checkpoint. Queues will build-up when the demand rate (arrival rate λ) exceeds the service rate (μ). The queue will stop increasing or decrease when the demand rate is equal to or lower then the service rate (see de Neufville and Odoni (2013) for appropriate models). Therefore, the arrival rate of the initial time interval, where $\lambda > \mu$, has an effect on the queue time build-up in the same time interval but also on the queue time experienced in future time intervals. Hence, there is a delayed effect between the arrival rate (measured at the entrance of the security checkpoint queue) and the security checkpoint queue time.

In addition to the delayed effect based on traditional queueing theory, a delay is encountered between the airport arrival time and the security checkpoint queue arrival time. This delay is caused by the fact that the passengers that are for example not checked-in online will have to pass the check-in facility. In addition, the goodbye area introduced in chapter 4 could also increase the time passengers spent between entering the airport and arriving at the security checkpoint. An example of this delay is shown in fig. 6.2.

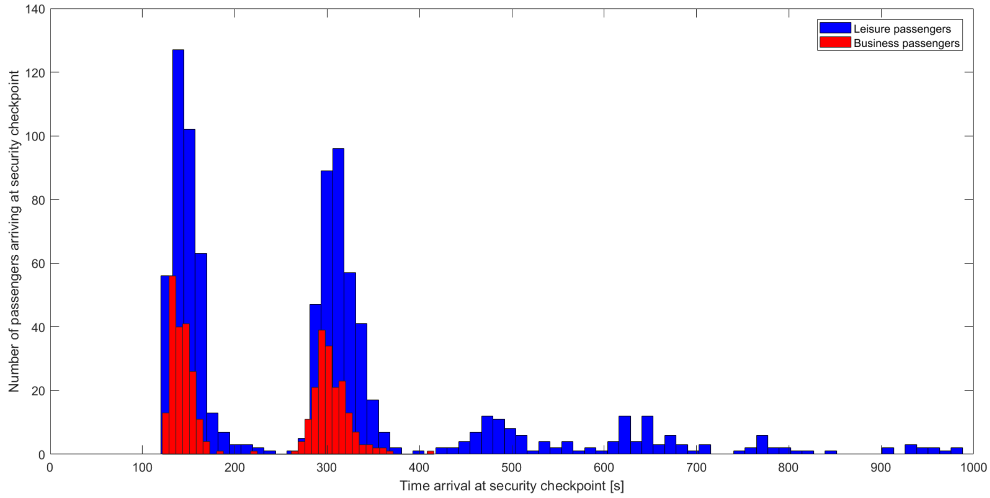


Figure 6.2: The time it takes to walk from entrance to the security checkpoint for business and leisure passengers.

In fig. 6.2 it can be seen that for both business and leisure passengers there are two peaks. The first peak (for business and leisure), mostly represent people walking directly towards the security checkpoint. Therefore, the time to the security checkpoint is the lowest. The second peak are the people (business and leisure) that need to pass by the check-in counter. Both peaks show that business passengers are on average just a little bit faster. This is the result of the implemented walking speed differences.

The leisure passenger data shows additional peaks. These peaks are a result of the implemented autonomous decision making process. Note, that due to the autonomous decision making, the latter part of the graph is specific for a certain scenario. For example, if only small flights depart from a certain gate, more people might decide to visit the goodbye area and stay for a longer period. Whereas in busy scenarios, passengers tend to visit the goodbye area less often and the amount spent in the goodbye area fluctuates more heavily. This fluctuation is caused by quick changes in queue length.

These delays could be one of the reasons that there is a lagged response in the $SCQT$. As a result the $SCQT_i^C$ (the security checkpoint C queue time in time interval i) is not only dependent on the people arriving in the same time interval, but also on people arriving in earlier time interval. Hence, it is dependent on the current arrival rate (AR_i) of passengers and the arrival rates of earlier intervals (AR_{i-l}). The number of lags l taken as input variables was determined based on the observations from graphs like fig. 6.2. It was observed that from all scenarios the longest time passengers needed to get from the entrance to their designated security checkpoint was around 25 minutes. This equals five time intervals ($l = 5$), therefore AR_i , AR_{i-1} , AR_{i-2} , AR_{i-3} , AR_{i-4} , AR_{i-5} were the lagged arrival rates taken into consideration.

The chosen parameters are not the only parameters that could be selected to create meta-models.

Parameters that reflect agent behaviour are for example not taken into account, but could be used to construct meta-models as well. These parameters could for example include the business/leisure ratio per flight and the carry-on/checked luggage ratio per flight. However, in the current study these parameters are only used as experimental setting.

6.2. Data generation

In this section the data generation process is discussed. This is a time consuming but very important step in the meta-modelling process. It determines the design space of the meta-models. Therefore, data should be gathered such that it is rich in diversity.

The design space is the region over which the meta-model is supposed to hold. Outside this region, the meta-model has a poor performance. If the design space is too small, then the meta-model is not general for the agent-based model. However, if it is tried to make the design space large, then this might result in unnecessary simulation runs that take a long time. As an example, one might want to design the model such that every scenario is covered. This means that there could also be a scenario in which only large flights (>350 passengers) depart from Schengen gates. In reality this does not occur that often at e.g. AAS (see table 4.2), and these scenario runs are therefore unnecessary.

The data gathering simulations were therefore bound by two aspects. First, the size of aircraft were bound by the aircraft sizes present at AAS (shown in table 4.2). Secondly, the turn around time assumption (assumption 4) restricts the amounts of flights that can be assigned to the gates and therefore drastically reduces the design space.

In the reference papers used (Miyoung and Goel (2000), Miyoung Shin et al. (2014)), the data gathered was spread evenly across the design region. It was decided that the same approach would be adopted for this data gathering step. Initially, the aircraft that could depart from the Schengen and non-Schengen gates were divided into groups according to their size. This can be seen in table 6.1.

Table 6.1: Grouping aircraft according to the aircraft size into four groups.

	Aircraft size (#seats)			
	≤100	(100,200]	(200,300]	>300
Aircraft size group	S	M	L	XL

To create a rich data set, first the extreme scenarios were run. The simulation length was set to seven hours. This meant that in an extreme case only four XL aircraft are able to depart from a non-Schengen gate (taking assumption 4 into account). In the other assumed extreme case, only small flights would depart from the non-Schengen gate. The design area in between the extremes was filled with runs of different combinations of flight sizes and amount of flights. An example set of the experiments that were performed to gather data are shown in table 6.2. In this table only the experiments are shown for gates A and C. The experiments conducted for gate B and D are equal to gate A and C, respectively.

What should be noted from the table is that the extreme case with four XL flights at a non-Schengen gate was not performed, because the agent-based model and simulations ran into problems when that amount of passengers arrived at the fictitious airport. For example, if the queues at the security checkpoints became too large, passengers could get pushed into a corner by the other waiting passengers. The victims would then not be able to escape that corner any more, which resulted in bad output data. In addition, in some realisations the JAVA code completely stopped due to encountered errors.

In these simulation experiments there can be two types of systems present (Charnes (1993)). There can be *Terminating systems* in which the model has specific start-up and shut-down times and there can be a *steady-state system*. However, the simulation is a repetition of terminating systems (aircraft arriving/departing), that can result in a steady-state system. Furthermore, the parameters of interest are the security queue times for each time interval. Recall, that with the time interval standard assumption

Table 6.2: Subset of the experiments carried out for data gathering in terms of flight sequencing and aircraft size for gate A and C. Standard aircraft sizes taken were: (S = 100 pax), (M = 200 pax), (L = 300), (XL = 490).

	Flight sequencing														X-ray scanners
No.	Gate A							Gate C							per SC
1	S	S	S	S	S	S		S	S	S	S	S	S		2
2	S	S	S	S	S	S		S	S	S	S	S	S		3
3	XL	L	XL	M	M	L		L	M	M	L	M	M		2
4	XL	L	XL	M	M	L		L	M	M	L	M	M		3
5	L	M	S	XL	L	L		L	M	L	M	M	L		2
6	L	M	S	XL	L	L		L	M	L	M	M	L		3
7	M	M	M	M	M	M	M	M	M	M	M	M	M	M	2
8	M	M	M	M	M	M	M	M	M	M	M	M	M	M	3
9	XL	M	S	M	M			XL	M	S	M	M			2
10	XL	M	S	M	M			XL	M	S	M	M			3
11	M	XL	M	XL	M			M	S	M	M	M	M		2
12	M	XL	M	XL	M			M	S	M	M	M	M		3
13	S	XL	L	S				M	M	M	M	S	S		2
14	S	XL	L	S				M	M	M	M	S	S		3
15	L	XL	S	L				M	M	S	M	M	L		2
16	L	XL	S	L				M	M	S	M	M	L		3
17	XL	M	XL	M	M	XL		L	M	S	M	M	L		2
18	XL	M	XL	M	M	XL		L	M	S	M	M	L		3

(assumption 3) it was assumed that during a time interval the changes in queue time were minor. Thus, the queue time is treated as steady state in a time interval. Concluding, the current simulation data cannot be clearly identified as a terminating nor steady-state system. Therefore, it was decided to take an approach that fits both types of data analysis. This is done by first splitting the data into three phases. The main reason for the split of the data will become clear in section 6.3, which has to do with properties of the data set.

1. *Start-up phase*: The beginning of the simulation where the first flight has not yet departed and the passenger stream of the second flight is not yet present.
2. *In-operation phase*: Main phase of the simulation, where flights are departing one after another.
3. *Terminating phase*: Final phase where queues are slowly declining and only the passenger stream of the last flight is present.

The three phase periods are determined using the outputs of many simulations. During these simulations, the first flight was assigned three hours (simulation hours) after simulation initialisation. The entire planning horizon was filled with flights with minimal turn-around times.

The end of the start-up phase was determined every simulation run by applying a threshold rule:

$$t_{end,phase\ I} = \max\ t_{int}\ ;\ SCQT_{int} < \frac{1}{3} \times \max\ (SCQT)$$

This threshold was set using knowledge gained about the queue time behaviour, such that a stationary process is present in phase II. For example in fig. 6.3, it can be seen that after interval number 20 the threshold was crossed. Hence, in this example it was decided that the start-up phase ended at interval number 20. On average the start-up phases ended at interval number 21, leaving out simulations where no queue build-up occurred.

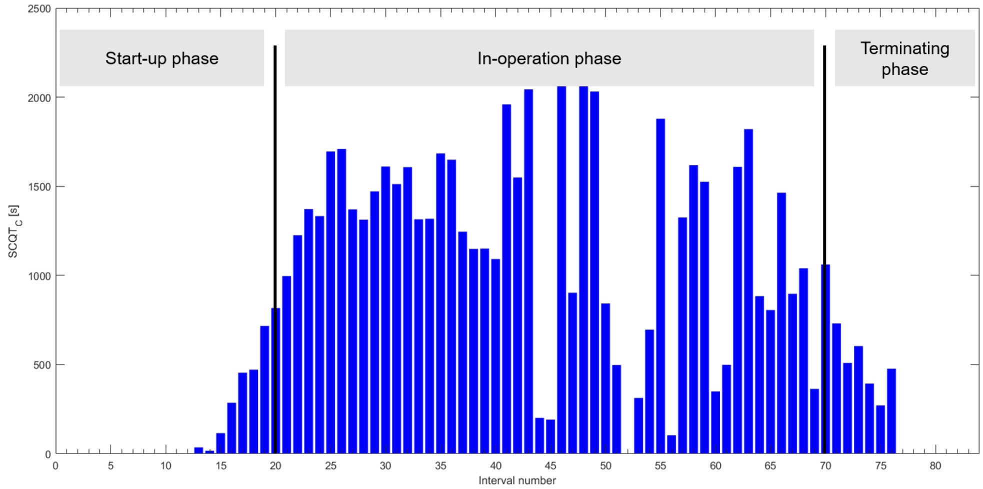


Figure 6.3: The three phases displayed in a sample scenario.

Therefore the global end of the start-up phase (phase I) was set to time interval 21 (105 minutes after initialisation). Furthermore, it was observed from all experiments that the queue time would only start to build-up from interval eleven onwards. Therefore, the meta-models should be built such that they start forecasting from interval eleven. Hence, the initial ten time intervals were discarded.

The rationale used to determine the terminating phase was based on the arrival streams of passengers. In the final 70 minutes before the last flight, only passengers for the last flight (that departs at the end of the simulation) arrive at the entrance of the airport. Since the minimum turn around time was set to 40 minutes (assumption 4) and in the final thirty minutes no passengers arrive any more (see fig. A.1). Hence, the final 14 time intervals (70 minutes) are designated as terminating phase (phase III), which is also shown in fig. 6.3. The time in between the start-up and terminating phase is the in-operation phase (phase II).

After determining the three distinct phases in the data set, the batch mean method was applied (Charnes (1993)). This method "attempts to deal with autocorrelation in the data by combining adjacent auto-correlated observations in the output sequence into (nearly) uncorrelated batches". This method was basically already applied when it was decided to create the time intervals. In addition, the method of independent replications with truncation was applied. The simulation is run several times (ten times) starting from the same initial conditions, but using independent pseudo-random numbers for each replication. Hence, every replication is independent of one another.

Besides gathering the output data of all the security checkpoints during the experiments, the input data was also gathered. After controlling for outliers, the output observations were coupled to the input arrival rates (e.g. $AR^A_{i-l}, l = 0, 1 \ldots, 5$). The resulting data set was of the form displayed in table 6.3.

Table 6.3: Columns of the created data set.

AR_{i-5}	AR_{i-4}	AR_{i-3}	AR_{i-2}	AR_{i-1}	AR_i	Interval no	Phase	Gate	$SCQT_i$
…	…	…	…	…	…	…	…	…	…

Finally, the data was split per gate and also per amount of X-ray scanners active. This resulted in a total of 24 data sets, for which individual meta-models had to be built. In fig. 6.4 the dataset split into

24 separate data sets is shown.

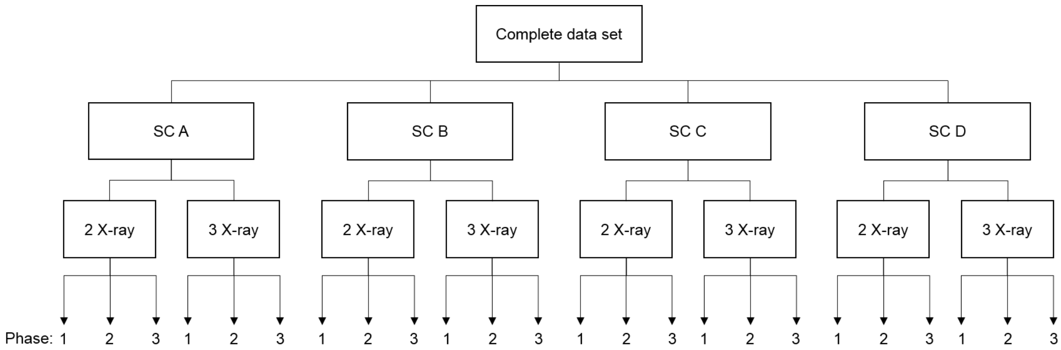


Figure 6.4: The split of the dataset per checkpoint, number of X-ray scanners and phase.

6.3. Polynomial response methodology

In this section the first type of meta-modelling, a form of response surface methodology, will be explained and created. First the form of the model will be determined by examining the data type and the relationships among the input/output variables. What will be seen is that multi-collinearity is apparent among the variables which will have to be reduced, before model fitting can take place. Finally, the importance of the parameters is determined and the performance of the models will be tested. The steps taken in this chapter are taken from the book by Woolridge (2012).

The polynomial response methodology tries to approximate the response in some region of the dependent variables by a polynomial model (regression). The approach fits first or second order polynomial models to the responses y. The common form of such a polynomial regression is shown in eq. (6.1). Where p_k is a polynomial with inputs $\mathbf{x}$. $f(\mathbf{x})$ is the predicted output response.

$$f(\mathbf{x}) = \sum \beta_k p_k(\mathbf{x}) \tag{6.1}$$

For the remainder of this chapter $\mathbf{y} = (y_1, \dots, y_n)'$ represents a set of outputs of the simulation model ran under input conditions $(\mathbf{x_1}, \dots, \mathbf{x_n})$, respectively. The ϵ_i for the multiple observations are assumed to be independent, identically distributed quantities with variance σ^2.

First, the data needs to be examined in detail, started off by recalling the purpose of a meta-model. The aim of the meta-model is to forecast the security checkpoints queue times per time interval on an arbitrary day. This means that the model needs to be general. The goal is to perform a flight-to-gate assignment optimisation for a planning horizon that is least half-a-day long. However, the simulations ran were only seven hours. This has some complications on the methods that can be applied.

In fact, the data collected is panel data. As normally in panel data, data is gathered from different groups of e.g. individuals over time. The time-series dimension of the current data are the consecutive time-interval observations. The conventional method to apply would be panel data regression techniques. An example of a panel data regression is shown in eq. (6.2). Letting i denote the cross-sectional unit and t the time period, we can write a model with a single observed explanatory variable as

$$y_{it} = \beta_0 + \delta_0 d2_t + \beta_1 x_{ij} + a_i + u_{it},\ t = 1, 2. \tag{6.2}$$

In the case of the current dataset i denotes the different scenarios and t denotes the time interval. The variable $d2_t$ is a dummy variable that equals zero when $t = 1$ and one when $t = 2$. This dummy variable does not change across individuals, therefore it does not have a subscript i. In the example, there only exist two time intervals, whereas in the data there exist more than two. The panel data method applied to the current problem, would use a different dummy variable for each time interval, except one. This is a disadvantage because it means that the model can only be used to forecast the security checkpoint queue time for the same amount of time intervals as was used to determine the estimators. Furthermore, it is restricted to forecast per specific individuals (this case seven hours). Hence, there is no such thing as out of sample forecasting which is necessary in the current research.

Therefore, an autoregressive distributed lag model is proposed. Which is mainly employed in time-series data. In the current study, a special type of time-series data is available. The obtained data from the individual simulations are time-series that are not connected over time. Such that the complete data set is a collection of small time-series. The autoregressive distributed lag estimation performance best when there is a certain level of stationarity in the output variable, which was artificially created in the experiment design phase. This is achieved in current study by creating waves of queue times (see fig. 6.3) and by splitting the data set in the three phases. The result was that the probability distribution of the queue time was stable over the time intervals, as approximately the same realisation can be found in different time intervals.

In addition, it was observed that the dependent variable tends to be larger when the previous time interval observation of the dependent variable is also larger. Therefore, a partial autocorrelation graph was made of the dependent variable, security checkpoint queue time. An example of such a graph is shown in fig. 6.5.

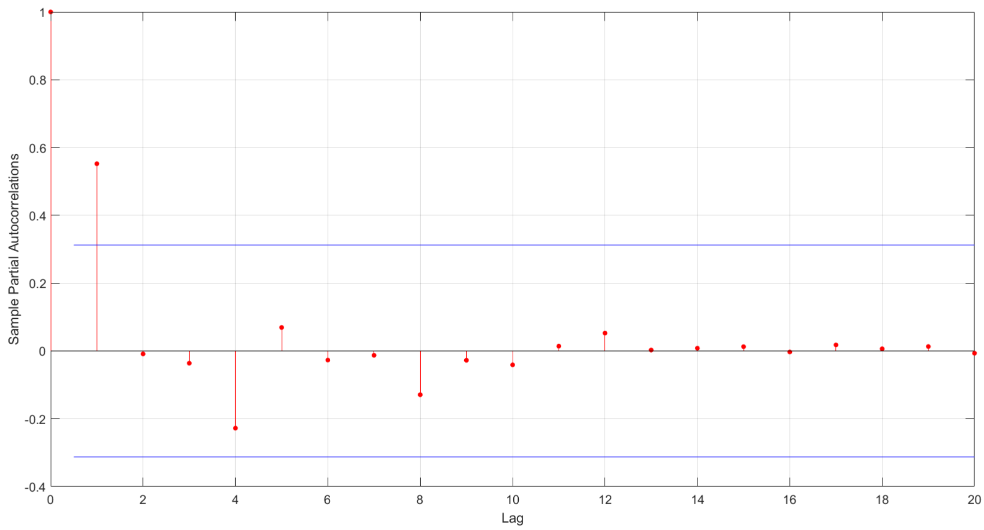


Figure 6.5: The partial correlation graph of the dependent variable $SCQT^C$.

From this figure it can be observed that the security checkpoint queue time has significant auto correlation at lag 1. Therefore, it was decided to include the autoregressive term y_{i-1} (moving average of order 1). However, one should be careful that no serial correlation is present in the error terms, which therefore needs to be checked. The chosen model is shown in eq. (6.3).

$$SCQT_i = \alpha_0 + \alpha_1 SCQT_{i-1} + \sum_{l=0}^{5} \beta_l IAT_{i-l} + u_i \tag{6.3}$$

where i indicates the time interval, l the lag and IAT is the inter-arrival time of passengers during the time interval, $SCQT_i$ and IAT_{i-l} are stationary processes and u_i a white noise process. The IAT was chosen because it was already in the same units as the dependent variable (seconds). Finally, this means that an extra column was added to the data set displayed in table 6.3, for the observed security checkpoint queue time at lag 1 ($SCQT_{i-1}$). Furthermore, the previously specified AR terms, needed to be converted to IAT terms.

Before estimating the coefficients or estimators by using ordinary least squares (OLS), one should be sure that the assumptions for OLS to be consistent and efficient hold. These time series Gauss-Markov assumptions are mentioned below.

- TS.1: Linear in parameters,
- TS.2: No perfect collinearity,
- TS.3: Zero conditional mean,
- TS.4: Homoskedasticity,
- TS.5: No serial correlation.

The mathematical definition of all assumptions will not be elaborated upon here, but an exhaustive explanation can be found in Woolridge (2012). Theory tells us that OLS is consistent and unbiased conditional on $\mathbf{X}$ (all explanatory variables) under assumptions TS.1 - TS.3. However, under assumptions TS.1 - TS.5 the OLS standard errors, t statistics and F statistics hold. The most important assumptions to control for in the current study are assumptions TS.2 and TS.5, because the others will hold or can be checked for once these assumptions hold. TS.5 has been partially accounted for by including the lagged dependent variable $SCQT_{i-1}$. However, TS.5 still needs to be checked, as well as TS.2.

6.3.1. Controlling for perfect collinearity

To check for perfect collinearity one is interested if one explanatory variable can be perfectly explained by a linear combination of the other explanatory variables. If this would be the case then OLS would be inconsistent. A check for collinearity can be done by using a pairplot (Zuur et al. (2007)). This plot is a scatterplot matrix, that shows all pairwise scatter plots of all variables in one graph. If perfect collinearity is apparent then one of the scatter plots shows a perfect linear relationship between one of the explanatory variables. Note, that only perfect collinearity would result in inconsistent OLS estimates. On the other hand if multicollinearity would be detected, then the interpretation of the coefficients would be not possible. Therefore, also in the case of non-perfect collinearity some extra steps had to be performed.

The first pairplot was created from one of the most demanding data sets: security checkpoint C with only two operators active.

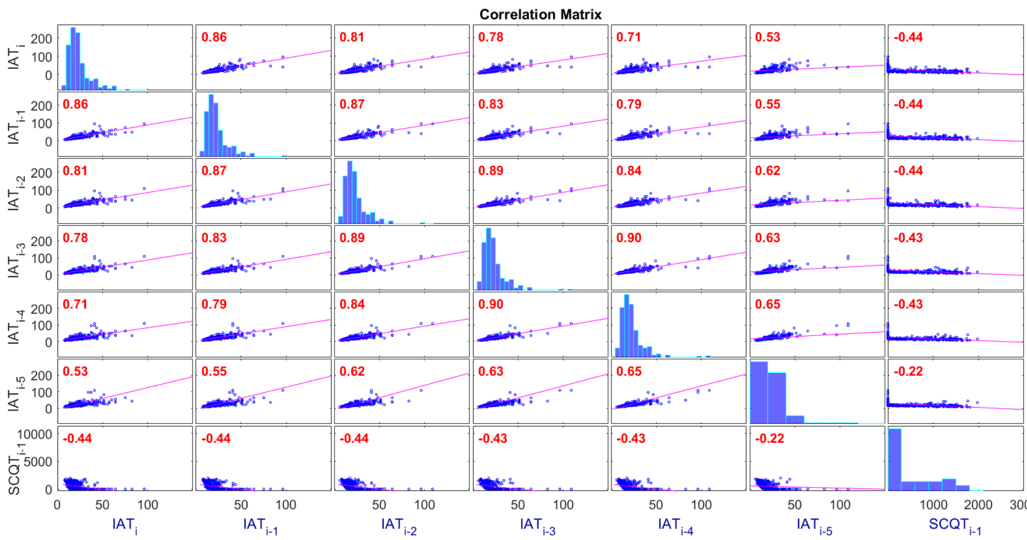


Figure 6.6: Pairplot of the explanatory variables to detect (perfect) collinearity, with in red the R^2 values.

What should be observed in this plot-matrix are the relatively high correlation values. The graph shows high correlation values, but there is no sign of perfect collinearity (correlation equal to one). However, high correlations mean that there is multicollinearity. Multicollinearity makes it difficult to uncover the partial effect of each variable on the dependent variable. A R^2 value of 0.9 between x_1 and x_2 as example means that 90% of the sample variation in x_1 can be explained by the other explanatory variable x_2. There is not much that can be done to remove collinearity, except for dropping variables or combining variables to reduce the multicollinearity (Woolridge (2012)). Hence, it was decided to use combined variables in the fitting stage.

Another pairplot was made of all variables (explanatory as well as dependent variables). This pairplot could indicate if there are explanatory variables that are clearly linearly related to the independent variable. This pairplot is shown in fig. 6.7.

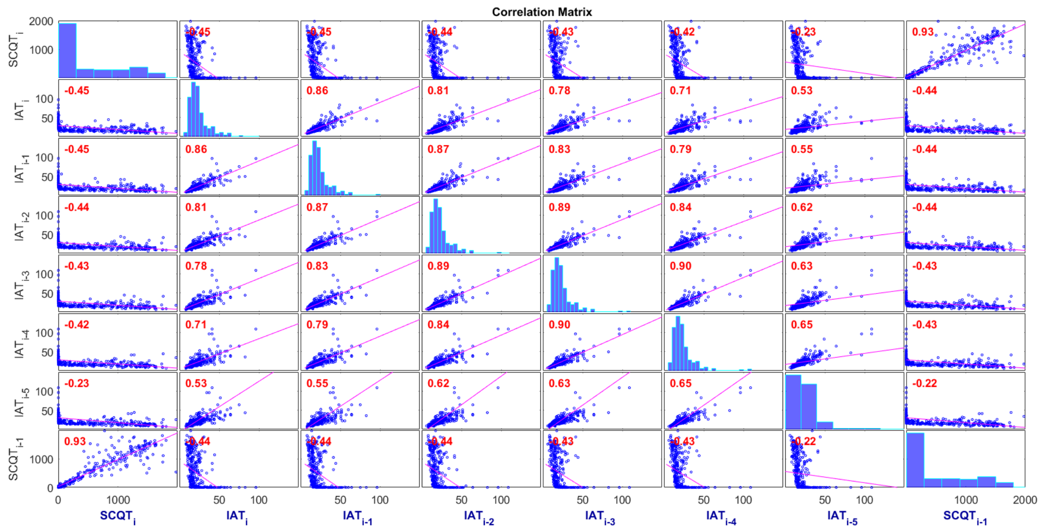


Figure 6.7: Pairplot of all variables, with in red the R^2 values.

This pairplot shows that there are no clear linear relationships between the IAT_{i-j} terms and the dependent variable. However, there is most probably a form of linear relationship between the $SCQT_{i-1}$ and the $SCQT_i$. This was expected since fig. 6.5 already showed this correlation. In addition, this dependent variable provides a simple way to account for historical factors that cause current differences in the dependent variable that are difficult to account for in other ways.

6.3.2. Fitting the model

The fitting stage involves two steps. In the first step the efficiency of the model is improved. The collinearity is removed by stepwise adding and removing explanatory variables and combinations of the explanatory variables. In the second step the fitted model is tested against a validation data set. This data set is different from the one used for fitting and will give a indication of the general performance of the model in out of sample situations.

A model will be made for each of the 24 data sets: Each security checkpoint (SC A, B, C, D), per amount of X-ray scanners (2 or 3 X-ray scanners) and per phase (3 phases). Ideally, the fitting of the models would be done per data set, but this is a tedious process and does not add to the academic value of this thesis. Hence, it was decided to fit the models and thus combine the explanatory variables to reduce the collinearity on the data sets for two X-ray scanners (assumption 5). The same variables and linear combinations of the variables would be used for the three X-ray scanner data sets. Hence, the following assumption:

Assumption 5 (Removing collinearity simplification) *It is assumed that the data obtained by employing 2 X-ray scanners behaves in a similar way as data obtained by employing 3 X-ray scanners. Hence, the explanatory variable combinations that reduce collinearity in the 2 X-ray scanner data set are the same for the 3 X-ray scanner data set.*

These combinatorial variables were obtained by performing a step-wise linear regression. The criterion to add or remove terms was the p-value for an F-test of the change in the sum of the squared error by removing or adding a term. If the decrease in the sum of squared errors was insignificant, showing a p-value higher than 0.05 (5% significance level), then it was removed.

The starting point is the normal linear regression on all the explanatory variables, see table 6.4.

Table 6.4: Estimated coefficients of a linear regression of the explanatory variables on $SCQT_i^C$, phase 2 and two X-ray scanners. The $R^2_{adj} = 0.8660$ and the root mean squared error $RMSE = 203\ s$.

	Estimate	SE	tStat	pValue
(Intercept)	92.068	31.474	2.9252	0.0036464
IAT_i	-0.53213	1.6843	-0.31594	0.75222
IAT_{i-1}	-1.9121	1.9736	-0.96883	0.33324
IAT_{i-2}	1.883	2.0167	0.93371	0.35104
IAT_{i-3}	-2.39	2.0961	-1.1402	0.2549
IAT_{i-4}	0.64681	1.756	0.36834	0.71282
IAT_{i-5}	0.15254	0.4373	0.34882	0.72742
$SCQT_{i-1}$	0.90417	0.021034	42.985	7.99E-149

This table shows once more that there exists collinearity among the variables, because almost all p-values are insignificant but the adjusted R^2_{adj} value is high. Therefore, the stepwise linear regression method was employed to reduce the multicollinearity. The selected variables and combination of variables by the stepwise regression method are shown in table 6.5.

Table 6.5: Stepwise linear regression result for SC C phase 2 with two X-ray scanners.

	Estimate	SE	tStat	pValue
(Intercept)	82.509	28.749	2.87	0.004331
IAT_{i-1}	-1.9573	1.0359	-1.8895	0.059571
IAT_{i-5}	0.1759	0.38421	0.45783	0.64733
$SCQT_{i-1}$	0.98949	0.03742	26.443	9.46E-89
$IAT_{i-5} \cdot SCQT_{i-1}$	-0.00526	0.00195	-2.6972	0.007298

To double check that there is no multicollinearity apparent, the Belsley collinearity test was performed on the selected variables (IAT_{i-1}, IAT_{i-5}, $SCQT_{i-1}$, $IAT_{i-5} \cdot SCQT_{i-1}$). This test is one of the most common used tests for collinearity (Belsley et al. (2005)) and has been included in table B.1 as a formality only. It showed no clear evidence of harmful collinearity.

The next step is to analyse the residuals of this regression, because it was assumed that there was no serial correlation present. In order to check for serial correlation, the residuals were acquired from the regression above. Since the data set has been made from multiple runs, the time series data has gaps in them. Therefore, the serial correlation should only be checked within one run and not between runs (or for the entire data set all at once). To check for serial correlation in the residuals, the following check was performed to see if there is significant evidence of serial correlation:

$$Corr(u_i, u_s) = 0\ \forall i \neq s,$$

where u_i are the residuals in time interval(period) i. An example of this test for the regression above is shown below:

$$Corr(u_i, u_{i-1}) = 0.3$$

which showed the presence of autocorrelation or serial correlation in the residuals. As this was unexpected, a second graph was made of the residuals, which is shown in fig. 6.8. What can be seen is that the residuals are not a white noise process.

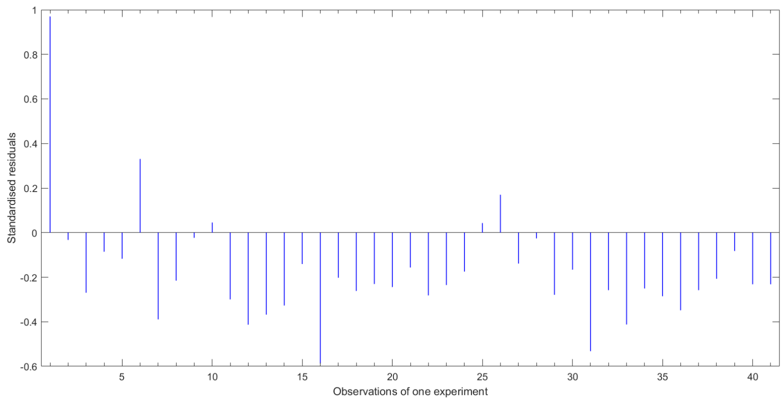


Figure 6.8: Residual plot created using data of a single experiment, which shows the absence of a white noise process.

This meant not only that assumption TS. 5 was violated but also TS.3 did not hold. Hence, the p-values used to construct the above relationship were unreliable.

Therefore, it was decided to perform random sampling such that the data set was converted from a panel data set to a cross-sectional data set. Each random sample contained the variables listed in eq. (6.3). Random sampling made sure that the data set only consisted out of independently sampled observations. This means that from every experiment random samples of data were drawn, which formed the new data set. Hence, the resulting data consists only of random observations of the dependent variable with their corresponding explanatory variables, as a randomly pooled cross-sectional data set. Therefore, by definition no serial correlation will be present any more (Woolridge (2012)).

Once these new data sets were established, previous steps had to be repeated. The resulting stepwise regression is shown in table 6.6.

It can be seen that the fit is rather high with $R^2_{adj} = 0.906$, however the root-mean squared error is high ($RMSE = 171\ s$). There are multiple reasons for the bad fit of the regression meta-model including:

- The regression model form is not able to approximate the non-linear agent-based model data.
- The selected independent variables are not able to approximate the response.
- The data split into three phases is still too rough.
- The time-interval selection of five minutes could have affected the fitting errors.

Hence, if one wants to improve the quality of the regression meta-models, these are the most important areas to consider. In addition to the poor fit of the models, one can observe a high p-value for the IAT_{i-4} term (pValue = 0.7998). The stepwise regression has not decided to remove this term since the removal would not result in a significant increase in R^2_{adj} value. Also the variable was not linearly dependent with any of the other terms in the current model, since otherwise it would have been removed.

It was assessed that the residuals represented a white noise process (see appendix B, fig. B.2). Finally, the meta-models made for the $SCQT_i^C$ of security checkpoint C, phase I,II and III with two X-ray scanners are given in eqs. (6.4) to (6.6).

Table 6.6: Final linear regression on $SCQT_i$ of SC C, phase 2 and two X-ray scanners. The $R^2_{adj} = 0.906$ and the root mean squared error $RMSE = 171\ s$.

	Estimate	SE	tStat	pValue
(Intercept)	256.21	82.924	3.0897	0.002239
IAT_{i-1}	-5.7531	2.7251	-2.1112	0.035788
IAT_{i-3}	-7.5947	3.2207	-2.3581	0.019167
IAT_{i-4}	0.53927	2.1243	0.25386	0.79982
IAT_{i-5}	-1.2944	0.76359	-1.6952	0.091327
$SCQT_{i-1}$	0.97856	0.053793	18.191	4.00E-47
$IAT_{i-1} \cdot IAT_{i-3}$	0.17447	0.074156	2.3528	0.019437
$IAT_{i-4} \cdot SCQT_{i-1}$	-0.01614	0.004085	-3.9518	0.000102
$IAT_{i-5} \cdot SCQT_{i-1}$	0.009492	0.003838	2.473	0.014088

$$SCQT_i^{C,1,2} = 7.75 + 1.16\ SCQT_{i-1} \tag{6.4}$$

$$\begin{aligned} SCQT_i^{C,2,2} = 256.21 - 5.75\ IAT_{i-1} - 7.59\ IAT_{i-3} + 0.54\ IAT_{i-4} - 1.29\ IAT_{i-5} + 0.98\ SCQT_{i-1} \dots \\ + 0.17\ IAT_{i-1} \cdot IAT_{i-3} - 0.016\ IAT_{i-4} \cdot SCQT_{i-1} + 0.0095\ IAT_{i-5} \cdot SCQT_{i-1} \end{aligned} \tag{6.5}$$

$$SCQT_i^{C,3,2} = -2.49 + 0.85\ SCQT_{i-1} \tag{6.6}$$

where the superscript $C, 2, 2$ means that it is made for security checkpoint C, phase 2 and with two X-ray scanners, respectively. As was mentioned in assumption 5, the same form of the models will be used to fit against the $SCQT_i^{C,1,3}$,$SCQT_i^{C,2,3}$ $SCQT_i^{C,3,3}$ data sets.

Equations (6.4) and (6.6) show that in phase I and phase III the only variables that are able to say something about the dependent variable is the lagged dependent variable itself. Hence, the data obtained in these phases behaves like a random walk. This is unfortunate since the strictly exogenous variables in these phases are unused.

What should be observed from eq. (6.5), is the relationship between the inter-arrival times and the queue time. The coefficients of IAT_{i-1}, IAT_{i-3} and IAT_{i-5} are negative. Hence, there is a negative relationship between the inter-arrival time and the security checkpoint queue time. This was expected because if the amount of passengers arriving per interval is higher (IAT lower) then the queue time increases and vice versa. This result is also one of the few ways to test the validity of the meta-models to represent reality. All of the variables mentioned above are also in as a combinatorial variable. Therefore, the overall relationship between the explanatory variables (e.g. IAT_{i-1}) and the dependent variables could be assessed using local sensitivity analysis. However, the relationship of these variables will be dependent on the values of the other variables. The combinatorial variables make inference difficult and therefore a global sensitivity analysis will be carried out below. In the data used these variables have been created because they proved to have a large explanatory power on the variance of the dependent variable. Furthermore, it can be seen that certain lagged inter-arrival times are missing. This is a result of the stepwise linear regression.

This process was repeated multiple times for all the data sets shown in fig. 6.4. The regression results for all meta-models are shown in appendix B in table B.2 to table B.22. Note, that the split of the models into phases and number of X-ray scanners has been successful, as the parameters and the coefficients are significantly different from one another. Furthermore, the common knowledge of queue behaviour has been properly captured: higher service rate reduces the experienced queue time when the arrival rate is similar. This can be seen in fig. B.4 as the meta-models created for three X-ray scanners active show significantly lower experienced queue times compared to the models for two X-ray scanners active.

6.3.3. Global sensitivity analysis

The above described model for $SCQT_i^{C,2,2}$ is dependent on many variables that add uncertainty to the output of the model. With global sensitivity one can investigate the uncertainty that the different parameters propagate to the output. The output of global sensitivity analysis shows which variable contributes to most to the uncertainty in the output.

For an exact explanation of the steps taken to assess the global sensitivity analysis, one is referred to Saltelli et al.. At this moment it is only important to know that the relative magnitude of S_{Ti} is a measure for the proportion of the uncertainty that has been propagated to the output by the particular parameter. Ranking the parameters from largest contributor (S_{Ti}) to lowest, one can clearly see the most important parameters. The results of the global sensitivity analysis on the model in eq. (6.5) is shown in table 6.7. In addition the results of the global sensitivity analysis of the meta-model for $SCQT^{B,2,2}$ is shown.

Table 6.7: Result of the global sensitivity analysis for $SCQT^{C,2,2}$ regression meta-model.

Parameter	S_{Ti}	Distribution
IAT_{i-1}	1.44	U(5,600)
IAT_{i-3}	1.39	U(5,600)
$SCQT_{i-1}$	0.031	U(0,2000)
IAT_{i-5}	0.0041	U(5,600)
IAT_{i-4}	0.0026	U(5,600)

Table 6.8: Result of GSA for $SCQT^{B,2,2}$ regression meta-model influencing parameters.

Parameter	S_{Ti}	Distribution
IAT_{i-5}	1.33	U(0,600)
IAT_{i-4}	1.20	U(0,600)
$SCQT_{i-1}$	0.11	U(0,2000)
IAT_i	0.063	U(0,600)
IAT_{i-3}	1.59E-02	U(0,600)

What can be seen is that for both models an inter-arrival time variable is the most important and third most important is the $SCQT_{i-1}$ variable. Remarkable is that for the $SCQT^{B,2,2}$ model, the inter arrival rate at lag 5 is most important and not the instantaneous inter-arrival time (IAT_i) which is present in that model. This could be due to the fact that passengers heading for SC B are more likely to visit the goodbye area compared to the passengers heading for SC C. The result is that the passengers spend more time dwelling around the airport before continuing towards the security checkpoint and therefore the higher lagged inter-arrival time variables are more important.

6.3.4. Performance of the meta-models

The performance of this first set of meta-models can be checked by assessing the fit with a different dataset and assessing the forecasting performance. Especially, the latter is important since the data set has been split in three different phases. Hence, if the forecast is not correct in the first phase it will most definitely not be correct in the second phase either.

When assessing the fit with a different dataset, one is interested in the size of the generalisation error. Put differently: how well is the model in estimating the dependent variable using a different data set than used for fitting? This unique dataset is generated when the data was randomly sampled to reduce serial correlation. The one half of the initial data set is used for fitting and the second half (validation dataset) was used to assess the generalisation error. The generalisation error gives an indication of the general ability of the models to estimate the security checkpoint queue times in a certain time interval. In addition, if the generalisation error is far from the fitting error, then the model might be overfitted. On the other hand, if the generalisation and the fitted errors are similar, the model could be valid in general sense.

First, the estimation errors on the validation set were calculated. The results are shown in table 6.9. The table shows that the errors of the regression on the validation set were generally higher than on the fitting data set. Especially the second phase model for SC C has a significantly higher RMSE, indicating overfitting. This might be problematic when this model is incorporated in the optimisation, because the queue time estimates could be inaccurate and therefore indicating an optimum that is not an optimum at all.

Table 6.9: Regression performance on validation set compared to the fitted residuals.

	$RMSE_{fitted}$ [s]	$RMSE_{gen}$ [s]
$SCQT^{C,1,2}$	41.1	42.15
$SCQT^{C,2,2}$	171.0	272.94
$SCQT^{C,3,2}$	53.8	2.49

Remarkable is the lower generalisation error in the third phase with respect to the fitting error. This is due to the fact that the validation data set for phase III only contained observations close to 0 s queue time. This results in the error more or less equal to the intercept. All the fitting versus generalisation errors for the entire set of models have been reported in appendix B in table B.25.

Next, the forecast performance of the models is checked. This will be the main feature of the model that will be used when integrating it in the gate assignment optimisation and therefore the performance needs to be acceptable. The forecast performance of the regression models are tested on an arbitrary schedule that is shown in table 6.10. From this schedule the inter-arrival times (IAT) of the passengers per SC are calculated and used as inputs for the models. Furthermore, the $SCQT_{i-1}$ for the tenth time interval is set equal to zero. As the forecast starts at interval eleven.

Table 6.10: Schedule used for testing forecast, showing the time of departures as well as the expected number of passengers per flight.

Gate A		**Gate B**		**Gate C**		**Gate D**	
Pax	Time [h:mm]	Pax	Time [h:mm]	Pax	Time [h:mm]	Pax	Time [h:mm]
130	03:00	130	03:00	300	03:00	300	03:00
470	04:20	470	04:20	110	03:40	110	03:40
280	05:40	280	05:40	190	04:20	190	04:20
200	06:20	200	06:20	280	05:40	280	05:40
510	07:40	510	07:40	210	07:00	210	07:00
				130	07:40	130	07:40

Furthermore, simulations are performed of the schedule above, which is used to compare the forecast with. The resulting data is different from the fitting data, because the schedule is slightly longer and the amount of passengers per flight are different.

In fig. 6.9 the forecast is shown for security checkpoint queue A and B using two X-ray scanners. The forecast errors are high with respect to the level of the realisations. This was as expected since the fitting and generalisation errors found were also high (table B.25). As explained earlier the phase I models are only dependent on the lagged dependent variable. This means that whatever schedule is used, the forecast will always have this shape and queue time at interval 21. This is a major flaw, because the realised security checkpoint queue times are not always equal to this value. However, this is the effect of averaging over all the different experiments. This indicates that the security checkpoint queue time is difficult to forecast in phase I with this type of model. Hence, this resulted in the predicted increase in phase I. Unfortunately, the realisation of the simulation did not show this increase. Furthermore, the time interval at which the predicted queue time increase in phase II is close to the time interval where the realisation increases (for both A and B). However, both forecasts predict a too high queue time peak around time interval 43. In general, it seems like the forecast is able to predict the time at which the queue time starts to increase correctly. However, the level of the queue time

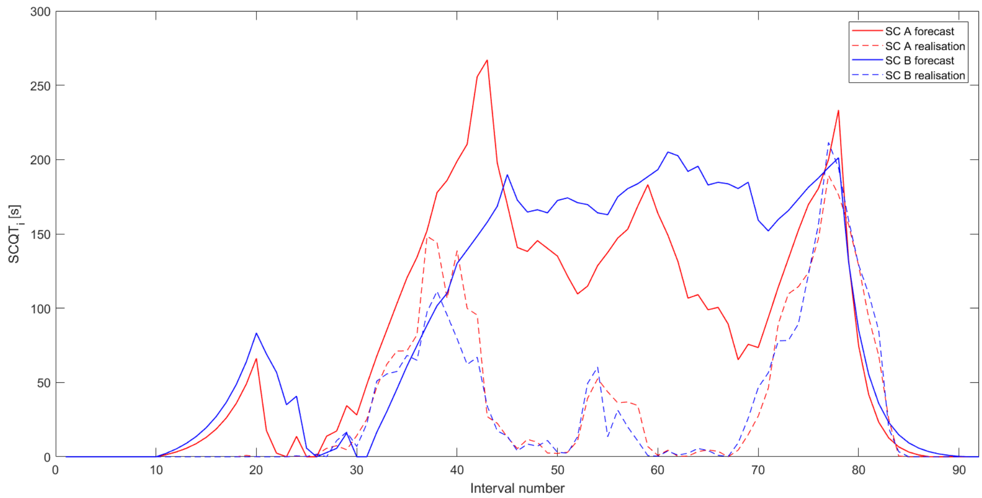


Figure 6.9: Security checkpoint A and B (2 X-ray scanners) queue time observed (realisation) and regression meta-model forecasts based on schedule in table 6.10 ($RMSE^A$ = 76.0 s , $RMSE^B$ = 96.6 s).

forecast is too high (in this scenario) compared to the realisation. Hence, the middle section (phase II) shows a consistent over-forecast. This can be caused by the fact that the linear regression model can only react in firm direct way on changes in the input. The final peak (around interval 79) is predicted well for both checkpoints.

The forecast produced for security checkpoint C and D with two X-ray scanners active is shown in fig. 6.10. Overall, the forecast of queue times is way higher than the realised queue times. The root

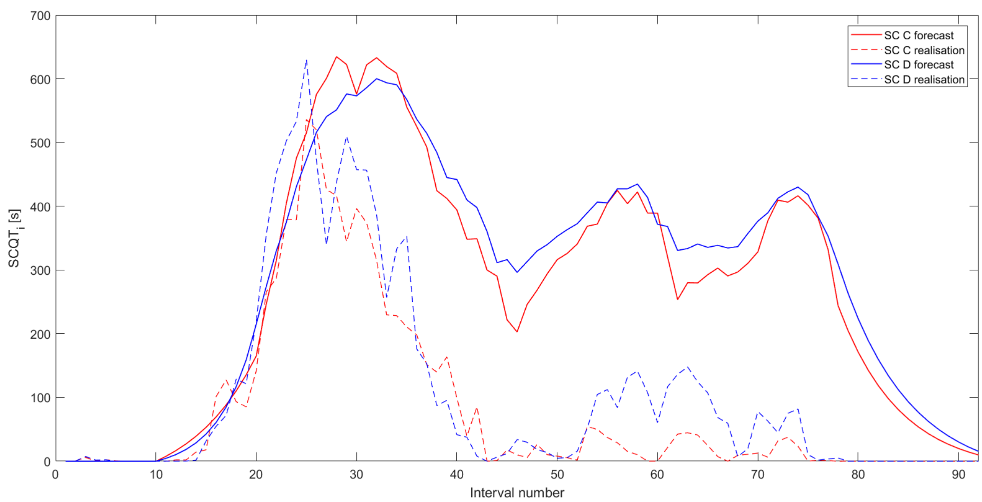


Figure 6.10: Security checkpoint C and D (2 X-ray scanners) queue time observed (realisation) and regression meta-model forecasts based on schedule in table 6.10 ($RMSE^C$ 238.4 s , $RMSE^D$ 308.8 s).

mean square error of the queue forecast for SC C was equal to 238 s ($\approx$ 4 min). That is high, knowing that the maximum realised queue time at checkpoint C is equal to 536 seconds. Furthermore, what can be seen is that the forecast in phase I (until interval number 21) is, as expected, only dependent on the lagged security checkpoint queue time. In the first eleven time intervals in fig. 6.10 there is zero queue time forecast as was explained earlier when introducing the phases. Fortunately in this case the realisation is close to the forecast in phase I.

Another observation is the fact that the first peak's shape of elevated queue time forecast seems to match the realisation. This reflects the fact that the phase II model has up to a certain level the ability to predict how fast the queue time increases in the beginning of the phase. The same was observed by predicting a 'busier schedule', which can be seen in fig. B.3. The meta-model has problems forecasting the rest of the in-operation phase. Figure 6.10 shows a clear overestimation in the second part of phase II. The forecasts clearly do not follow the shape of the realisations any more. The realisation reaches the first peak and then quickly reduces. However, the forecasts predict the initial peak to be longer. This introduces after the first peak the over-forecast, which continues till the end. In general, it cannot be said that this model is over-forecasting the security checkpoint queue times, since fig. B.3 shows a clear under-forecast. That is unfortunate because the difference between the forecasts in (figs. 6.10 and B.3), is then not equal to the difference in realisations between the two scenarios.

It can be concluded that these meta-models are able to capture partly the emergent behaviour in the agent-based model queues as the shape of the forecast show similar behaviour as the realisations. However, the model is not able to forecast the queue times accurately (RMSE = 171s). The forecast errors are too high with respect to the level of the realised queue times. Especially, if one aims to fulfil the IATA level of service standards, then the accuracy of the models is too low. These results show that the polynomial response or regression meta-models are not able to fully represent the agent-based model considering queue times.

The poor forecasting performance could be due to the fact that the average queue times per scenario vary quite heavily. In one scenario the maximum queue time is close to 40 minutes and in another the queue times are close to one. This is difficult for a model to generalise. Especially, when restricting the polynomial response meta-model to be linear (with linear combinations). Therefore, a non-parametric model would be the next step when looking for a valid meta-model for the agent-based model of an airport.

6.4. Gaussian radial basis function meta-model

In this section, a Gaussian radial basis function (GRBF) meta-model is built that should be able to replace the agent-based model in the simulation optimisation strategy. The objective is to construct a GRBF meta-model that approximates an unknown input-output mapping on the basis of given simulation data. Just as in the previous meta-model (Polynomial response methodology) the goal is not to provide an exact fit to the data but to develop a meta-model that captures the underlying relationship. It can be used to predict the output at some future observation of the input. This property is called the generalisation ability. As was discussed in chapter 2, a GRBF meta-model has been successfully applied to queueing systems in the work by Miyoung Shin et al. (2014). Of all types of meta-models available, this felt like the best candidate to explore. Furthermore, the steps taken by Miyoung Shin et al., will also be taken in the current study.

First the general idea of Gaussian radial basis function meta-modelling will be explained. Then, in section 6.4.1 many GRBF meta-models will be developed for the data sets created in section 6.2. The trade-off between fitting and generalisation errors will help selecting the final GRBF meta-models in section 6.4.2. In section 6.4.3, the importance of the parameters is assessed by global sensitivity analysis. The section is concluded by assessing the forecasting performance of the GRBF meta-models in section 6.4.4.

Suppose there is a given $n \times d$ input space input matrix $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \dots \mathbf{x_n}^T)$, where each of the n input vectors $\mathbf{x_i}, i = 1, \dots, n$ is in an d-dimensional space. $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$ is the target vector, whose elements are the individual output observations y_i corresponding to the input vector $\mathbf{x_i}$. The problem to be solved is the mapping from the d-dimensional input space to a one-dimensional output value based on the simulation data.

Finding the right meta-model involves trading off the overfitting versus underfitting. In general, the more variables used for fitting, the lower the fitting error (overfit). However, this model will not possess the ability to generalise on unseen data (data different from fitting dataset). This phenomena is the bias and variance dilemma which ideally looks like the graph in fig. 6.11. Models with a low complexity tend to have a high bias, while complex models have low bias but high variance.

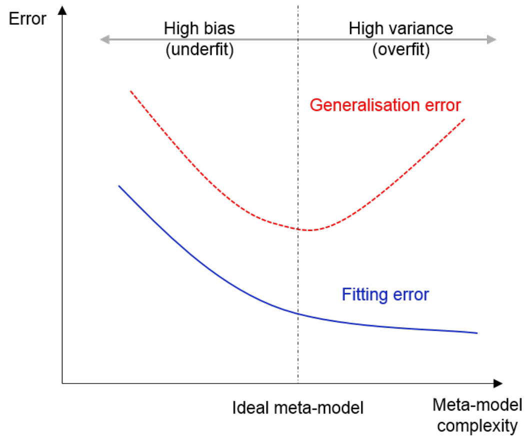


Figure 6.11: Idealised depiction of bias-variance and fitting and generalisation errors, taken from Miyoung Shin et al. (2014).

According to Mehrotra et al. (1997) artificial network meta-models can be evaluated by assessing the quality of the results, the generalisability and computational resources (or complexity). The quality of the results is reflected in the fitting error tested by the RMSE calculation. The generalisability is measured by assessing the error on the validation data set, that is different from the set used for developing the model. Finally, the computational resources are measured in terms of CPU time, memory requirements or training time. These are heavily dependent on the number of nodes, connections, and layers in a network.

Recall that the Gaussian radial basis function is of the form shown in eq. (6.7), which represents the mapping $f : R^d \rightarrow R$.

$$f(\mathbf{x}) = \sum_{j=1}^{m} w_j e^{\frac{-||\mathbf{x} - \boldsymbol{\mu}_\mathbf{j}||^2}{2\sigma_j^2}} \tag{6.7}$$

where $\mathbf{x} \in R^d$ is the input vector, $\boldsymbol{\mu}_\mathbf{j} \in R^d$ is the *j*th basis function centre, $||\cdot||$ denotes the Euclidean distance, w_j is the weight of the *j*th basis function and the σ_js are the basis function widths or spread. The function $f(\mathbf{x})$ will in the current research become for example become $SCQT^{C,2,2}$. The Gaussian radial basis functions play the role of transfer function in the neural network. The GRBF is completely defined by the parameters $(m, \boldsymbol{\sigma}, \boldsymbol{\mu}, \mathbf{w})$, where m is the number of basis functions with widths $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_m)$, basis function centres $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_m)$ and the weights from the basis functions to the outputs are $\mathbf{w} = (w_1, w_2, \ldots, w_m)$.

A GRBF model can also be seen as a three-layer-network. The first layer is the input layer that distributes the input vectors to each of the (m) nodes in the hidden layer, without any multiplicative factors. The m hidden units each represent a basis function and plays a role in performing nonlinear transformation of the input vector, producing a value as an output. In the third layer, the outputs from the m hidden units are linearly combined by using weights w to produce a single value model output.

This means that the problem of designing the GRBF involves choosing $3m$ parameters: m centres, m widths and m weights. However, in most research a global width is used $\sigma = \sigma_i \; \forall i = 1, \ldots, m$. Therefore, in the current research a global width will also be used, such that the total number of parameters to be determined reduces to $2m + 1$.

The method used to solve this problem is the Shin Goel (SG) algorithm, based on a mathematical framework for radial basis functions (Miyoung and Goel (2000), Shin and Goel (1998)). The SG algorithm is a three-stage process shown in fig. 6.12.

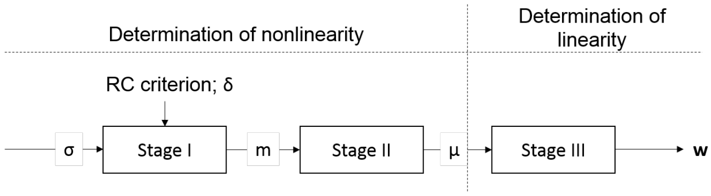


Figure 6.12: Meta-modelling steps as developed by Shin and Park (2000).

In this procedure, the nonlinear parameters m $\boldsymbol{\mu}$ and σ are determined, without reference to the output values. Once these parameters have been found and fixed, the linear parameters $\mathbf{w}$ are determined by using reference target outputs. In Stage I the representational capability (RC) algorithm will be used which is based on a mathematical framework developed in Shin (1998). The RC algorithm selects m for a given σ and a specified value of δ (the representational capability). In Stage II, for a given (m, σ) pair, the centres are determined and in the final stage the weights are determined by the least squares method. For the full mathematical basis of the RC algorithm one is referred to Shin (1998). Here only a brief conceptual description of the underlying theory is given.

The first question to ask in Stage I in fig. 6.12 is: For a given spread (σ), how many basis functions are needed to cover the input space adequately when it is known that 100% coverage can only be achieved by all the n inputs? To answer this question Shin (1998) introduces the representational capability (δ), which is defined relative to the input space spanned by the data $\mathbf{x}$. In the paper it was found that with $m \ll n$ still a very high representational capability could be obtained. Then in the Stage II the m centres of the n input vectors should be chosen such that the best design is created. This is done by choosing m vectors that are the farthest apart from each other, leading to a model having the maximum representational capability with m basis functions. Furthermore, selecting the centres in this way provides structural stabilisation, which is an important property of a good model.

6.4.1. GRBF meta-model development

In this subsection the GRBF models are created following the stages displayed in fig. 6.12. Before developing the models, it should be discussed which variables are used. Since radial basis function models are part of the non-parametric modelling one could argue that the steps taken in section 6.3 to remove collinearity are not needed since the model's performance is not diminished by the multi-collinearity problem of regression analysis (Wray et al. (1994)). However, more recent research focusses on the use of hybridised factor analysis-artificial neural networks (Garg and Tai (2012)). Hence, it was decided to follow this path and remove the multi-collinearity before applying an artificial neural network. The collinearity was removed, just as in section 6.3 by applying stepwise regression in advance. Fortunately, the stepwise regression to remove collinearity was already performed in section 6.3. Therefore, it was decided to use the same variables that are apparent in the regressions (eqs. (6.4) to (6.6) and table B.2 to table B.24). As an example, the steps in the meta-model development will be shown of the model for $SCQT^{C,2,2}$. Only the final models of the other data sets (shown in fig. 6.4) will be given.

The variables used to construct the GRBF meta-model for $SCQT^{C,2,2}$ are shown in table 6.11.
Furthermore, when applying a neural network the data must be normalised such that the individual features behave more or less like standard normally distributed data (Zuur et al. (2007)). So each data set in fig. 6.4 must be normalised. This is most commonly done by subtracting the mean value of each feature, and then scale it by dividing each feature by their standard deviation. An example of the normalisation of IAT_i is shown in eq. (6.8).

$$IAT_i^n = \frac{IAT_i - \mu_{IAT_i}}{\sigma_{IAT_i}} \tag{6.8}$$

Table 6.11: Variables used to create GRBF meta-model for $SCQT^{C,2,2}$.

$SCQT^{C,2,2}$ variables
IAT_{i-1}
IAT_{i-3}
IAT_{i-4}
IAT_{i-5}
$SCQT_{i-1}$
$IAT_{i-1}IAT_{i-3}$
$IAT_{i-4}SCQT_{i-1}$
$IAT_{i-5}SCQT_{i-1}$

where IAT_i^n is the normalised feature IAT_i, and μ_{IAT_i} and σ_{IAT_i} are the mean and standard deviation of feature IAT_i, respectively. These normalisation constants (mean per feature per data set, standard deviation per feature per data set) must be stored for forecasting purposes. Since, the input data that will be used for forecasting, needs to be normalised by the same constants.

As explained the GRBF meta-modelling steps involve fitting and assessing the generalisation error on different data sets. The split in data made was exactly the same as was used in the regression meta-modelling steps.

Step I: Selection of σ and δ
In the first step of the RC algorithm a range needs to be selected for σ and a value for the RC measure δ. In the reference papers, the value range of σ was set using a heuristic:

$$0 \leq \sigma \leq \sqrt{d/2}$$

where d is the number of input variables. Furthermore, δ is usually taken to be:

$$0.1\% \leq \delta \leq 1.0\%$$

It was decided that the model needed to have a 99% representational capability, $\delta = 0.1$. Since $d = 8$ for $SCQT^{C,2,2}$ (see table 6.11), the value range of σ lies between :

$$0 \leq \sigma \leq 2.$$

Multiple spreads will be taken to the next step, such that in the final step of the GRBF modelling the best model can be selected based on the fitting and generalisation errors.

Step II: Determining the number of centres m
The number of centres per σ are determined using an interpolation matrix for each σ. After constructing the matrix, its singular value decomposition (SVD) needs to be performed. The result is a diagonal matrix of decreasing singular values $s_1 \geq s_2 \geq \cdots \geq s_n \geq 0$. These singular values will give the number of centres from the following:

$$m = \max_{1 \leq i < n} i; s_{i+1} \leq s_1 \times \frac{\delta}{100} \tag{6.9}$$

where $(100 - \delta)$% is the chosen RC criterion. This results in a number of centres m per spread σ, (σ, m) pairs.

For the example data for $SCQT^{C,2,2}$, the number of input data points is 250 and there are eight input variables. The 250×250 interpolation matrix G is constructed by substituting the input vectors $\mathbf{x_i}$'s ($i = 1, \ldots, 250$) in the following:

$$G = \begin{bmatrix} \exp(-\frac{||\mathbf{x_1}-\mathbf{x_1}||^2}{2\sigma^2}) & \exp(-\frac{||\mathbf{x_1}-\mathbf{x_2}||^2}{2\sigma^2}) & \dots & \exp(-\frac{||\mathbf{x_1}-\mathbf{x_{250}}||^2}{2\sigma^2}) \\ \exp(-\frac{||\mathbf{x_2}-\mathbf{x_1}||^2}{2\sigma^2}) & \exp(-\frac{||\mathbf{x_2}-\mathbf{x_2}||^2}{2\sigma^2}) & \dots & \exp(-\frac{||\mathbf{x_2}-\mathbf{x_{250}}||^2}{2\sigma^2}) \\ \vdots & \vdots & \ddots & \vdots \\ \exp(-\frac{||\mathbf{x_{250}}-\mathbf{x_1}||^2}{2\sigma^2}) & \exp(-\frac{||\mathbf{x_{250}}-\mathbf{x_2}||^2}{2\sigma^2}) & \dots & \exp(-\frac{||\mathbf{x_{250}}-\mathbf{x_{250}}||^2}{2\sigma^2}) \end{bmatrix}.$$

For example if $\sigma = 1.9$, then the interpolation matrix G becomes:

$$G = \begin{bmatrix} 1 & 0.372 & 0.268 & 0.0138 & 0.952 & \dots & 0.107 \\ 0.372 & 1 & 0.971 & 0.103 & 0.549 & \dots & 0.661 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.107 & 0.661 & 0.774 & 0.371 & 0.187 & \dots & 1 \end{bmatrix}.$$

For each σ, G was computed and the SVD of G yields the diagonal matrix of singular values from which the number of centres m could be determined. The number of centres, for different values of spread are shown in table 6.12.

Table 6.12: Values of (σ, m) pairs for $\delta = 1\%$.

RBF no	σ	m
I	0.1	248
II	0.3	201
III	0.5	115
IV	0.7	73
V	0.9	49
VI	1.1	35
VII	1.3	28
VIII	1.5	19
IX	1.7	16
X	1.9	13

Step III: Determining the centre locations $\boldsymbol{\mu}$

The centre locations $\boldsymbol{\mu}$ are found by using the K-means clustering algorithm (Hartigan and Wong (1979)). This algorithm aims to divide the total number of data points (with N dimensions) into K clusters so that the within-cluster sum of squares is minimised. The K-means clustering algorithm minimises the distance between x_i and the closest centre $\boldsymbol{\mu_k}$. The first step is to split $x_1, \dots, x_d$ into clusters $S_1, \dots, S_K$ then:

$$\text{Minimise} \sum_{k=1}^{K} \sum_{x_d \in S_k} ||\mathbf{x_d} - \boldsymbol{\mu_k}||^2.$$

In the current research one should replace K with m since these are the number of centres (or clusters). This minimisation can be solved iteratively, which will not be shown here but can be found in the original work by Hartigan and Wong (1979).

The centre locations determined for the example ($\sigma = 1.9, m = 13$) pair are given in table 6.13.

These centre locations have been determined for all the (σ, m) pairs, but have not been shown here due to the space it would require. With the centre locations known, the design matrices of each GRBF

Table 6.13: Basis function centres for ($\sigma = 1.9, m = 13$) pair.

	IAT^n_{i-1}	IAT^n_{i-3}	IAT^n_{i-4}	IAT^n_{i-5}	$SCQT^n_{i-1}$	$IAT^n_{i-1}IAT^n_{i-3}$	$IAT^n_{i-4}SCQT^n_{i-1}$	$IAT^n_{i-5}SCQT^n_{i-1}$
μ_1	-0.459	-0.434	-0.447	-0.207	-0.041	-0.371	-0.013	-0.026
μ_2	0.219	0.247	0.297	0.066	-0.893	0.008	-0.796	-0.788
μ_3	5.835	6.612	6.568	15.001	-0.898	10.286	-0.803	-0.794
μ_4	-0.680	-0.705	-0.682	-0.284	1.675	-0.473	1.089	1.063
μ_5	1.177	1.334	1.409	0.284	-0.898	0.905	-0.803	-0.794
μ_6	2.453	3.060	2.996	0.933	-0.898	2.787	-0.803	-0.794
μ_7	-0.365	-0.329	-0.289	-0.121	-0.744	-0.326	-0.655	-0.641
μ_8	-0.735	-0.726	-0.759	-0.304	1.001	-0.480	0.474	0.477
μ_9	1.472	1.677	0.247	-0.045	0.917	1.241	1.804	1.464
μ_{10}	2.031	0.380	0.308	-0.008	-0.875	0.711	-0.774	-0.765
μ_{11}	0.172	0.062	-0.042	-0.044	0.897	-0.060	1.372	1.390
μ_{12}	0.080	0.445	0.448	0.108	1.663	0.028	3.276	3.234
μ_{13}	4.025	1.479	1.464	0.370	-0.898	2.551	-0.803	-0.794

could be determined. The design matrix Φ, is the matrix used to determine the weights for each of the basis functions in the next step. Φ is created as follows:

$$\Phi = \begin{bmatrix} \exp(-\frac{||\mathbf{x_1}-\boldsymbol{\mu}_\mathbf{1}||^2}{2\sigma^2} & \dots & \exp(-\frac{||\mathbf{x_1}-\boldsymbol{\mu}_\mathbf{m}||^2}{2\sigma^2} \\ \vdots & \ddots & \vdots \\ \exp(-\frac{||\mathbf{x_{250}}-\boldsymbol{\mu}_\mathbf{1}||^2}{2\sigma^2} & \dots & \exp(-\frac{||\mathbf{x_{250}}-\boldsymbol{\mu}_\mathbf{m}||^2}{2\sigma^2} \end{bmatrix}.$$

In the example of $\sigma = 1.9$ and $m = 13$ the design matrix would be (250×13). These design matrices are made for all (σ, m) pairs, but are not explicitly shown here.

Step IV: Determining the weights $\mathbf{w}$ and estimating $\mathbf{y}$

After calculating all the design matrices, the weights of the m basis functions should be calculated. Since the number of centres is not equal to the number of data points, this has to be done by the pseudo inverse method. This means that the weights are given by:

$$\mathbf{w} = \Phi^+\mathbf{y}, \tag{6.10}$$

where Φ^+ denotes the pseudo inverse of Φ, $\mathbf{y}$ is the observed output vector of size 250×1, and $\mathbf{w}$ is the weight vector of size $m \times 1$. The weights for the previously used example are shown in table 6.14.

Table 6.14: Listing of weights for example GRBF for $\delta = 1\%$.

RBF no	σ	m	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_10	w_11	w_12	w_13
X	1.9	13	-808.9	136.6	1.21E-12	1144.8	-89.6	2.0	48.2	1288.7	1271.0	40.7	-1012.5	724.0	-55.6

At this point all the parameters $(m, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{w})$ have been determined. The last step is to compute the output $\hat{SCQT}^{C,2,2}$ by using eq. (6.7). Thus, the fitted GRBF for model X ($m = 13$ and $\sigma = 1.9$) is shown in eq. (6.11).

$$
\begin{aligned}
S\hat{CQT}^{C,2,2} = & -808.9 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_1||^2}{2(1.9)^2} + 136.6 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_2||^2}{2(1.9)^2} + 1.2E - 12 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_3||^2}{2(1.9)^2} \\
& + 1144.8 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_4||^2}{2(1.9)^2} - 89.6 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_5||^2}{2(1.9)^2} + 2.0 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_6||^2}{2(1.9)^2} \\
& + 48.2 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_7||^2}{2(1.9)^2} + 1288.7 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_8||^2}{2(1.9)^2} + 1271.0 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_9||^2}{2(1.9)^2} \\
& + 40.7 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_{10}||^2}{2(1.9)^2} - 1012.5 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_{11}||^2}{2(1.9)^2} + 724.0 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_{12}||^2}{2(1.9)^2} \\
& - 55.6 \exp -\frac{||\mathbf{x^n} - \boldsymbol{\mu}_{13}||^2}{2(1.9)^2}
\end{aligned}
\tag{6.11}
$$

The output $S\hat{CQT}^{C,2,2}$ is computed by substituting the normalised inputs into the above equation. Since this model needs an eight dimensional input (eight input variables), it is difficult to show the response behaviour in a single graph. Though, the meta-models created for phases I and III for all the security checkpoints only had one input variable ($SCQT_{i-1}$). Therefore, by means of an example the response of $S\hat{CQT}^{C,1,2}$ is given in fig. 6.13.

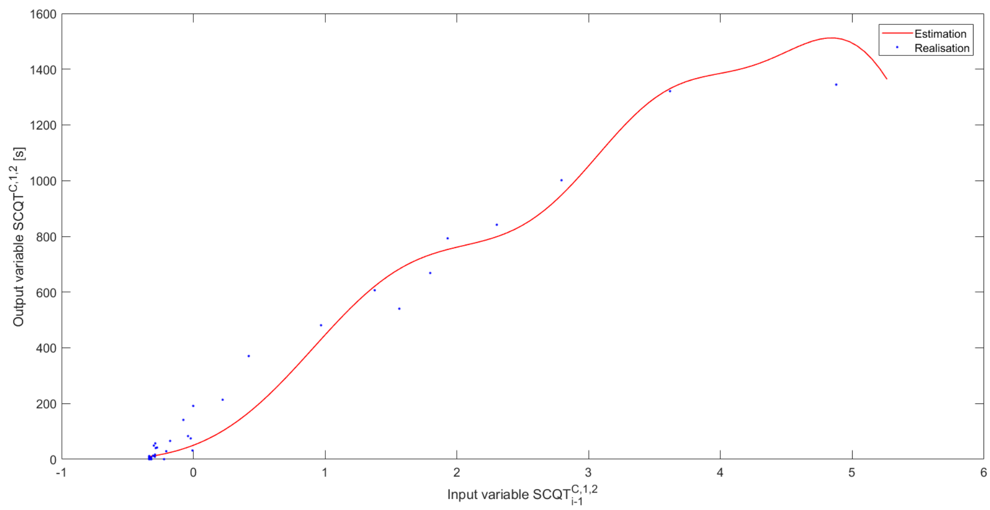


Figure 6.13: Fitted Gaussian radial basis function for $SCQT^{C,1,2}$ ($\sigma = 0.7, m = 5$).

In the figure, the simulated data is shown (realisation), together with the estimated data for the security checkpoint queue time of gate C in phase I with two X-ray scanners active. In the figure it can be clearly seen that there are five centres (five points from which the GRBF radially decay) with a relatively large spread (since the $0 \leq \sigma \leq \sqrt{2}$). In addition, it can be seen that the fit around zero is not very satisfying. Since the input variables have been normalised, this is the area where one wants to have a good fit with the data.

After determining all the radial basis functions for all the data sets, as well as the different (σ, m) pairs, the final GRBF model per data set needed to be selected.

6.4.2. Model selection

As explained earlier, the final GRBF models are selected based on the approximation ability, the generalisation ability and (if these are non determining) the complexity. Similar to the regression model evaluation, the approximation ability or fitting error and the generalisation error are evaluated using the root-mean-square error. In this section, the considerations will be mentioned for selecting the final GRBF meta-model for $SCQT^{C,2,2}$. Similar considerations held when selecting the other final GRBF meta-models.

The fitting and generalisation errors for the candidate models for $SCQT^{C,2,2}$ are shown in table 6.15. Furthermore, in fig. 6.14 the fitting and generalisation errors are plotted for the potential models in table 6.15. What can be clearly observed is that the fitting errors decrease with increasing m. This was expected since a large number of basis functions would provide a better approximation compared to only a couple of basis functions.

Table 6.15: Fitting and generalisation errors for potential GRBF models for $SCQT^{C,2,2}$.

	(σ, m) pairs		RMSE [s]	
RBF no	σ	m	Fitting	Generalisation
I	0.1	248	0.34	638.75
II	0.3	201	43.84	431.04
III	0.5	115	94.43	342.61
IV	0.7	73	126.96	309.33
V	0.9	49	138.13	302.28
VI	1.1	35	139.06	279.03
VII	1.3	28	146.44	268.87
VIII	1.5	19	175.99	283.36
IX	1.7	16	183.94	273.04
X	1.9	13	180.80	257.89

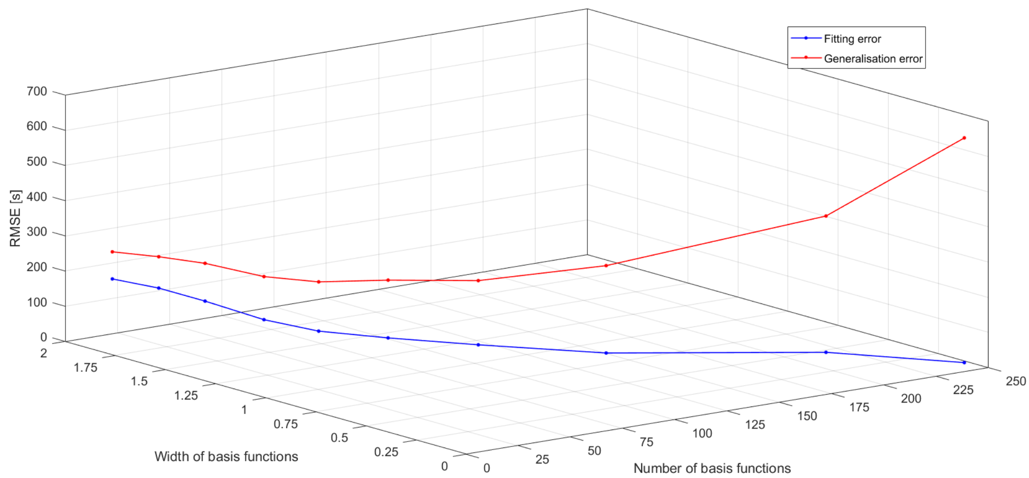


Figure 6.14: Fitting and generalisation errors plotted for GRBFs of table 6.15.

Choosing the final meta-model for $SCQT^{C,2,2}$ solely based on the fitting error, then one would choose model I. However as one can see in the table, the generalisation error of this model is the worst of all other models. Since the aim of the research is to find meta-models that are able to replace the agent-based model and simulation for the security checkpoint queueing, the generalisability should be leading. What can be observed in table 6.15, is a decline in generalisation errors goes with a decrease in number of basis functions. This was also anticipated due to the bias-variance trade-off. From model I to model IV the generalisation error shows the steepest decline, which then flattens out towards the minimum generalisation error of model X. However, the fitting error shows the largest increase from model I to model IV. Finally, it was decided that the generalisation error should be leading when selecting the final model, since the model will have to be used in many different situations (schedules). However, if the fitting error deteriorates stronger at higher values of spread, then a model might be considered with a slightly higher generalisation error. Hence, based on this analysis, the final GRBF model for $SCQT^{C,2,2}$ will be model X with $m = 13$ and $\sigma = 1.9$.

Similar as was decided in the removing collinearity simplification assumption (assumption 5), the spread σ selected for the two X-ray scanner models will also be selected for the three X-ray scanner models. So, since the GRBF selected for $SCQT^{C,2,2}$ contained basis functions with width $\sigma = 1.9$, the GRBF for $SCQT^{C,2,3}$ will also have basis functions with width $\sigma = 1.9$. The number of basis functions can differ between these two models.

The final chosen (σ, m) pairs as well as the fitting and generalisation errors of all meta-models are shown in table 6.16.

Table 6.16: Final selection of GRBF models for all data sets, showing the selected σ, m and fitting and generalisation errors of the final models.

	(σ, m) pairs		RMSE [s]	
GRBF	σ	m	Fitting	Generalisation
$SCQT^{A,1,2}$	0.5	4	9.93	95.67
$SCQT^{A,2,2}$	1.5	20	79.06	202.05
$SCQT^{A,3,2}$	0.5	5	0.69	0.48
$SCQT^{A,1,3}$	0.5	4	0.86	1.66
$SCQT^{A,2,3}$	1.5	15	17.35	18.24
$SCQT^{A,3,3}$	0.5	2	0.00	0.00
$SCQT^{B,1,2}$	1.1	4	23.37	17.82
$SCQT^{B,2,2}$	1.7	17	118.92	101.58
$SCQT^{B,3,2}$	0.5	6	1.57	0.02
$SCQT^{B,1,3}$	1.1	4	2.96	1.82
$SCQT^{B,2,3}$	1.7	11	10.94	28.94
$SCQT^{B,3,3}$	0.5	3	0.02	0.00
$SCQT^{C,1,2}$	1.3	4	34.31	28.77
$SCQT^{C,2,2}$	1.9	13	180.80	257.89
$SCQT^{C,3,2}$	0.3	10	56.68	60.97
$SCQT^{C,1,3}$	1.3	4	14.18	13.43
$SCQT^{C,2,3}$	1.9	10	39.52	88.75
$SCQT^{C,3,3}$	0.3	4	0.55	0.05
$SCQT^{D,1,2}$	0.9	4	38.11	19.80
$SCQT^{D,2,2}$	1.3	21	124.31	103.14
$SCQT^{D,3,2}$	0.9	4	108.27	83.88
$SCQT^{D,1,3}$	0.9	5	8.36	10.39
$SCQT^{D,2,3}$	1.3	15	56.86	66.75
$SCQT^{D,3,3}$	0.9	3	11.99	37.80

What can be seen from this table is that in general the generalisation errors of the meta-models made for the in-operation phase are higher than the other meta-models. Especially in the case that only two X-ray scanners were active. This could be, just as was the case in the regression meta-models, due to the fact that the output variable fluctuation was heavier in this phase with two X-ray scanners. The security queue time could be really high (e.g. 40 minutes queue time) but also relatively low (e.g. 1 minutes queue time). Hence, the maximum and minimum queue times could be really far apart in the second phase. Furthermore, the poor fit of the meta-models could again be caused by similar factors as were mentioned in section 6.3.2.

In table 6.17, an initial comparison of the meta-models made for $SCQT^C$ (regression and GRBF) is made.

Table 6.17: Comparison of the regression and Gaussian radial basis function meta-models made of $SCQT^C$.

	GRBF		**Regression**	
	RMSE [s]		RMSE [s]	
Meta-model	Fitting	Gener.	Fitting	Gener.
$SCQT^{C,1,2}$	34.31	28.77	41.10	42.15
$SCQT^{C,2,2}$	180.80	257.89	171.00	272.94
$SCQT^{C,3,2}$	56.68	60.97	53.80	2.49

What can be seen is that overall the GRBF meta-models show both a better fit as well as lower generalisation errors. Note, that both errors for both models in the second phase are rather high. This means that if there are time intervals where the realised queue time is around 1 minute, the meta-models could be off by around 4 minutes. These errors are too high for these less busy time intervals. On the other hand, if the queue time was around 30 minutes, then the 4 minutes off is not that big of an issue.

The aim of the split in data between the amount of X-ray scanners was aimed to split the data in high queue time data and low queue time data. However, this attempt did not fully split the data in these two halves. Hence, future research could focus on splitting the data in a different way. There could be looked at input value patterns that result consistently in really high queue times, medium queue times and low queue times. Then the data could be split into three sets, one with low queue time data, one with medium queue time data and the last data set with only high queue times. This could result in better performing meta-models.

6.4.3. Global sensitivity analysis

Just as was done for the phase II regression meta-models, a global sensitivity analysis is also performed for the GRBF meta-models. Recall, that one can investigate the uncertainty that the different parameters propagate to the output. The results of the global sensitivity analysis of GRBF models created for $SCQT^{C,2,2}$ and $SCQT^{B,2,2}$ are shown in tables 6.18 and 6.19.

Table 6.18: Result of GSA for $SCQT^{C,2,2}$ GRBF meta-model, showing the ranking of influencing parameters.

Parameter	S_{Ti}	Distribution
$SCQT_{i-1}$	0.488011	N(0,1)
IAT_{i-4}	0.030256	N(0,1)
IAT_{i-5}	0.019579	N(0,1)
IAT_{i-1}	0.008784	N(0,1)
IAT_{i-3}	0.005856	N(0,1)

Table 6.19: Result of GSA for $SCQT^{B,2,2}$ GRBF meta-model, showing the ranking of influencing parameters.

Parameter	S_{Ti}	Distribution
$SCQT_{i-1}$	0.602512	N(0,1)
IAT_i	0.064913	N(0,1)
IAT_{i-5}	0.002552	N(0,1)
IAT_{i-4}	0.001253	N(0,1)
IAT_{i-3}	3.57E-04	N(0,1)

What can be seen are different results compared to the GSA of the regression meta-models. The most important factor (which propagates the most uncertainty to the output) is the $SCQT_{i-1}$. Comparing the results of GSA for $SCQT^{B,2,2}$ to the GSA results of the regression model, one can see that the IAT_{i-5} is no longer the most nor second most important factor. Furthermore, when comparing the GSAs of the $SCQT^{C,2,2}$ meta-models one can see that in the GRBF meta-models, the inter-arrival times at lags 4 and 5 are the second and third most important factors. The comparison of both GSAs clearly indicate the difference the two types of models produced.

6.4.4. Forecast performance

Finally, the forecasting performance of the meta-models will be examined. This is done in exactly the same way as was done in section 6.3 and of the same schedule as presented in table 6.10.
In fig. 6.15 the forecast of $SCQT^A$ and $SCQT^B$ and the realised queue times with two X-ray scanners active are shown.

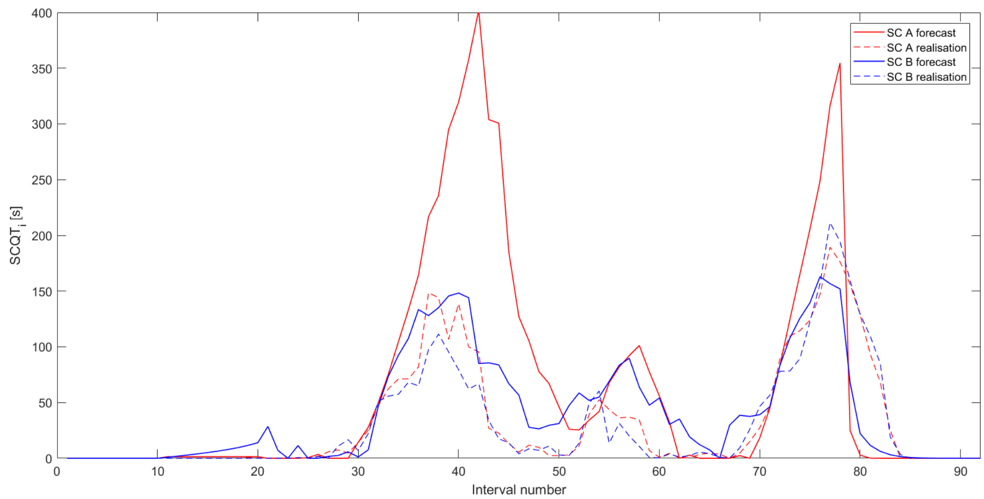


Figure 6.15: Security checkpoint A and B(2 X-ray scanners) queue time observed (realisation) and GRBF meta-model forecast based on schedule in table 6.10 ($RMSE^A$ = 81.2 s, $RMSE^B$ = 34.6 s).

What can be seen is that the graphs forecasts produced by the GRBF meta-models follow the realisations quite well. The models predict increases and decreases when they are actually realised. Especially, the meta-model created for SC B shows impressive results. The meta-models for SC A slightly over-forecast the initial and final peak. This probably caused the high forecast error ($RMSE^A$ = 81.2 s). However, the forecast's shape still shows actual queue behaviour in an acceptable way. In general, the forecast produced by the GRBF meta-models for $SCQT^A$ and $SCQT^B$ approximate reality better than the regression meta-models. Furthermore, in fig. C.1 the forecasts of the same scenario using the three X-ray scanner models are shown, which reflects common queue behaviour (lower queue times experienced when the service rate increases).

The produced forecast of $SCQT^C$ and $SCQT^D$ and the realised queue times with two X-ray scanners are shown in fig. 6.16.

Interesting to see is that the forecast produced by the GRBF meta-models for $SCQT^C$ outperform the forecast produced by the regression meta-models (fig. 6.10, RMSE = 235.0 s). Both type of models show more or less the same behaviour, but the regression models tend to over-react or is more sensitive to changes in the arrival rates (inter-arrival times). On the other hand the forecast produced for SC D by the GRBF shows a very high over-forecast. The model is not able to predict the level nor the shape of the real queue times. This is reflected by the higher forecast errors compared to the regression model forecast of $SCQT^D$. This could be due to the fact that the selected centre locations are not properly placed to forecast the security checkpoint queue time in this scenario, since no deviant

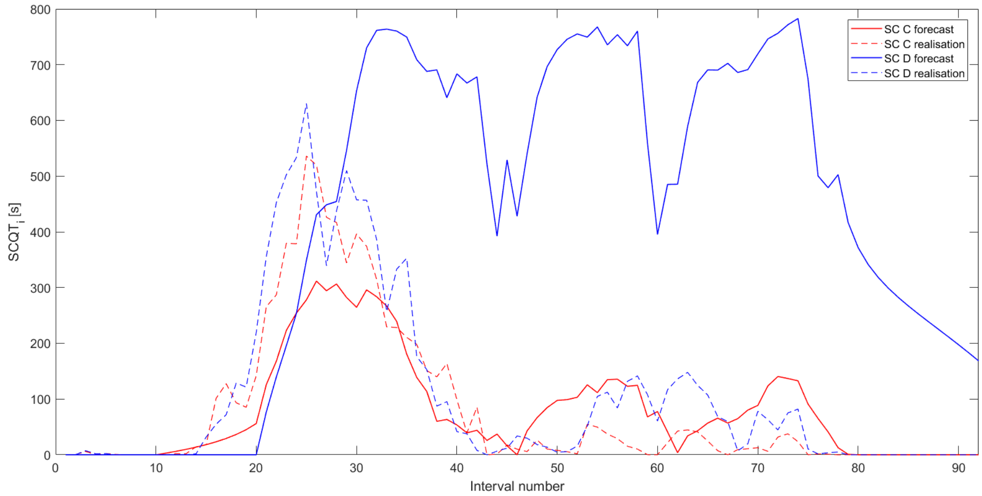


Figure 6.16: Security checkpoint C and D(2 X-ray scanners) queue time observed (realisation) and GRBF meta-model forecast based on schedule in table 6.10 ($RMSE^C$ = 72.6 s, $RMSE^D$ = 438.4 s).

behaviour was discovered in the model selection step (see table C.1). Future research could try to improve the performance of the model by examining the forecast performance, rather than examining a generalisation error. This could result in models with better forecast performance.

In appendix C fig. C.2, an additional forecast is shown of $SCQT^C$, for a different schedule, equal to the schedule used to produce fig. B.3. What can be observed is that the GRBF meta-model produces a worse forecast than the regression meta-models based on $RMSE$ (RMSE is 580 s and 779 s for regression and GRBF, respectively). In addition, the GRBF models do not follow the shape of the realisation as good as the regression models do. Hence, in a busy scenario, where queue times could reach up to 30 minutes, the GRBF meta-models perform worse than the regression meta-models.

More forecast examples and comparison between the two types of meta-models will be given in chapter 7.

6.5. Integration with differential evolution algorithm

Since the meta-models (both regression and GRBF) have been selected, the final step is to integrate the meta-models with the differential evolution algorithm. This section describes the method of integration, which will be done in a similar way as has been done in chapter 5. Specifically, the problem formulation given in section 5.1 remains unchanged, but the optimisation flow changes.

The major change compared to the simulation optimisation method, is the way the security checkpoint queue times are estimated, which are used in the objective function value with eqs. (5.2) and (5.3). In the simulation optimisation functional flow, the security checkpoint queue times were the output of the agent-based model and simulation. However, this method is very time consuming. A meta-model based optimisation could drastically decrease the computational time, at the cost of losing details of the governing agent-based model. These meta-models would replace the agent-based model and simulation, in the optimisation strategy.

In the proposed method in this section, the differential evolution algorithm can be used as it is designed. The functional flow diagram of the meta-model based optimisation is shown in fig. 6.17.

The trial vectors that are the output of the crossover process, will be used as input to the objective function module. Recall that it was assumed that that the number of passengers that will depart per flight, and their distribution of arrival is known in advance (assumption 2).

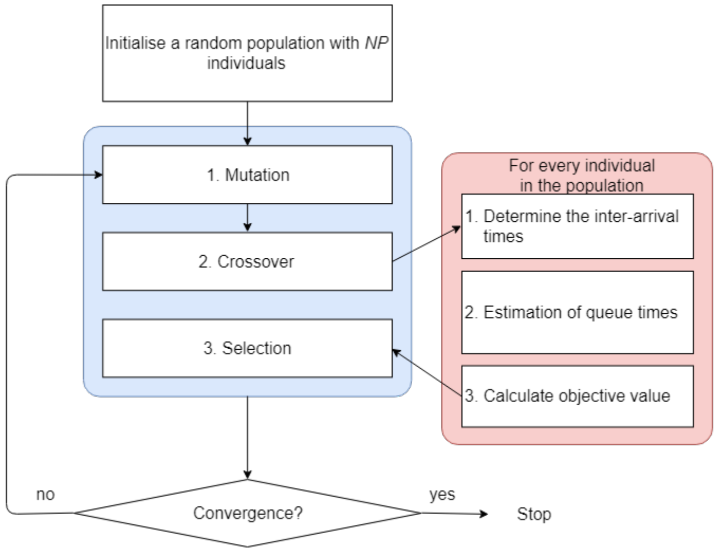


Figure 6.17: Functional flow diagram of the meta-model based optimisation framework.(Blue) Differential evolution algorithm. (Red) the objective function module.

The first step in the objective function module is the conversion from the binary gate assignment decision variable to passenger streams. Since the total amount of passengers (including Business/Leisure and transfer ratio) is known, the passenger streams (number of passengers per time interval) per departing flight can be calculated. If the differential evolution algorithm decides to assign a certain flight to e.g. SC A, then the passenger stream for that flight is added to the passenger stream expected for security checkpoint A. This is graphically shown in fig. 6.18.

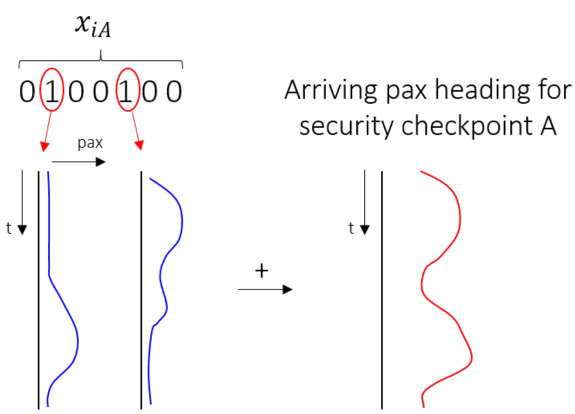


Figure 6.18: Step one in the objective function module shown in fig. 6.17.

After the number of passengers per time interval per security checkpoint have been determined, they can be converted into the IAT_i used in the meta-models. This is simply done by:

$$IAT_i = \frac{300}{PAX_i}$$

where the 300 represents the five minute time interval (in seconds) and PAX$_i$ the number of passengers that arrive in a five minute interval. The $SCQT$ for the entire horizon and all security checkpoints are calculated using the created meta-models in either section 6.3 or section 6.4. Before the inter-arrival times can be implemented in the GRBF meta-models, they need to be standardised with the same standardisation constants as were used to construct the meta-models. Furthermore, the decision variable r_j decides whether two or three X-ray scanners meta-models are used.

In the final step of the objective function model, the objective function values for all the individuals in the population are calculated using eq. (5.3). The way the security checkpoint queue times for every security checkpoint are calculated (step 2 in fig. 6.17) is graphically shown in fig. 6.19.

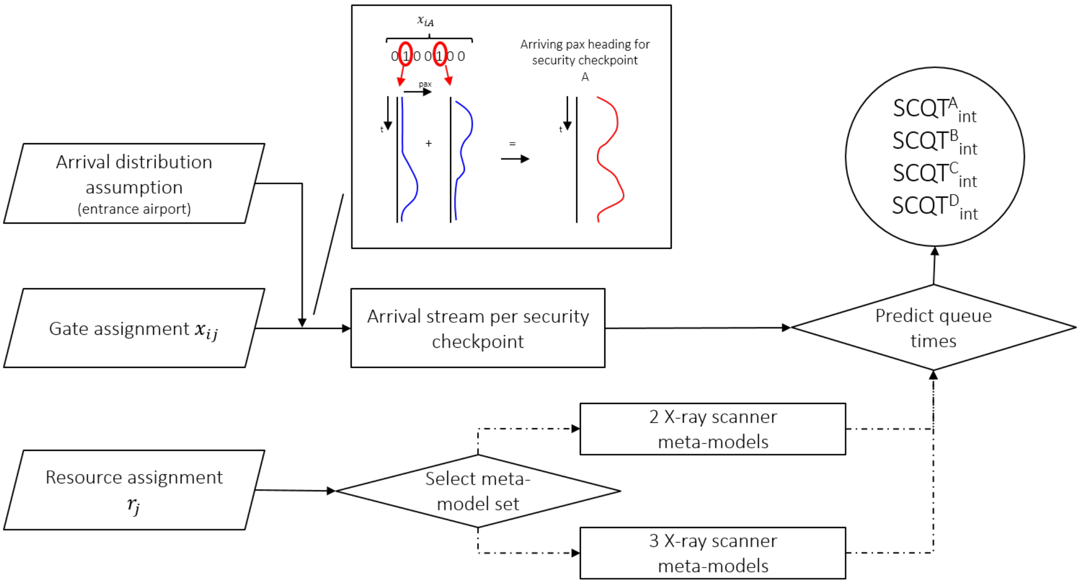


Figure 6.19: Objective function calculation based on the decision variables.

Note, that since the generalisation error of both the regression and GRBF meta-models were fairly high, it is expected that the optimum found by the optimisation will not reflect the actual observations from simulations properly. Therefore, the optimum found by this integrated optimisation method will need to be checked by performing simulations of the optimum decision variable found.

The final step is to assess the performance of the different ways of integration on test cases.

7

Evaluating methods of integration

In this chapter the different methods developed for integrating an agent-based model for airport passengers with a gate and resource assignment optimisation will be evaluated using two cases.

As described in section 3.6, a case study will be performed to assess the validity of the meta-model based integration by comparing the meta-model based integration with the direct integration method in section 7.1. Secondly, in section 7.2 a case will be used for verification, to study whether the proposed methods are able to find an optimal gate and resource allocation that minimises the differences in queue times amongst checkpoints.

7.1. Validation case study

In this section, the methods of the initial integration of an agent-based model with an operations research optimisation will be tested. Specifically, this case study is developed such that a comparison can be made between the three different methods of integration: Simulation optimisation, Regression model based optimisation, and GRBF model based optimisation.

This first test case is made such that the different methods can be fairly compared. It contains a small schedule, in which only two gates are used. The gate usage restriction was imposed due to the computational time that is needed by the simulation optimisation method to solve larger cases. As explained, the computational time needed for one simulation run depends on the size of the aircraft and the departure frequency. Secondly, the differential evolution algorithm explores both the feasible solution space and the infeasible solution space. Infeasible solutions can be very challenging since these can result in over two-hours-long simulation runs. In addition, the simulations are performed in JAVA, but the data handling and the differential evolution algorithm is done in MATLAB. This is a less preferred set-up, because the iterations involve time-consuming manual interaction.

Therefore, it was decided that the case study, developed to compare the integration methods, would be small and examines only the Schengen gates C and D (total number of gates $m = 2$). The small case schedule was developed based on knowledge about the observed behaviour during the research, which is shown in table 7.1. One can see, that the schedule only considers eight scheduled departures. Hence, the total number of flights (n) considered in the problem $n = 8$. Therefore, the total number of decision variables in this case study equals eighteen ($2 \times n$ plus the two X-ray scanner decision variables). Finally, the total number of X-ray scanners that need to be open is set to $s = 5$.

The settings used in the differential evolution algorithm are shown in table 7.2. These settings have been chosen such that the probability of convergence was the highest, by consulting the coder of this algorithm (PhD candidate Ho-Huu, V.) in addition to trial and error.

Constraint violation is increasingly penalised by the penalty setting, from one generation to another. In the first generation a solution that violates a constraint is penalised by the minimum penalty value, and

Table 7.1: Departure schedule for validation case study.

Flight	Pax	Time [h:mm]
1	150	3:00
2	150	4:20
3	200	5:40
4	250	7:00
5	300	3:00
6	200	4:20
7	210	5:40
8	300	7:00

Table 7.2: Differential evolution algorithm settings for validation case study.

Parameter	Value	Unit
Population size	8	Ind.
Stopping condition tolerance	1.00E-05	[-]
Total generations	30	It.
Penalty minimum	8	[-]
Penalty maximum	15	[-]

the final generation (Generation 30) by the maximum penalty value.

Since the validation case only examined two security checkpoint queues, the earlier defined objective function calculation in eq. (5.3) is cumbersome. Therefore, the following equation will serve as the objective function in this case study:

$$\min_{x_{ij}, r_j} f = \sum_{int=1}^{T} |SCQT_{int}^{C} - SCQT_{int}^{D}|. \tag{7.1}$$

7.1.1. Simulation optimisation

The simulation optimisation was the first method used to solve the flight-to-gate assignment problem of the schedule in table 7.1. The initial random generation had a population of eight individuals, of which none was a feasible solution. Each individual (ind) of the first generation are shown in table 7.3. Column one to eight of x_{iC} and x_{iD} represent the assignment variables of flights one to eight to gate C and D, respectively. Hence, if the x_{1C} is equal to one, then flight one is assigned to gate C.

As explained, the time needed to estimate these infeasible solutions was high. For example, a single simulation run of the second individual ($ind = 2$) in table 7.3 took a little over two hours. The differential evolution algorithm did not always converge towards a feasible solution, and even if a feasible solution was found it was checked whether this was a global optimum of local. Note, that in this report only the successful runs of the optimisation algorithm will be shown and the non-convergent runs are left out.

Table 7.3: Example of an initial population of decision variables (x_{ij})produced by the differential evolution algorithm.

ind	x_{iC}								x_{iD}								r_C	r_D
1	0	0	0	1	1	1	0	1	0	0	1	1	1	1	1	0	2	3
2	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	2	2
3	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	3	3
4	0	1	0	0	0	1	1	1	1	0	0	1	1	1	1	0	3	3
5	0	0	1	0	1	0	0	0	1	1	0	0	0	1	1	1	3	3
6	0	1	1	1	1	0	0	0	1	0	1	1	1	1	1	1	2	2
7	0	0	0	0	0	1	0	1	1	1	0	0	1	0	1	1	3	2
8	0	1	1	0	1	0	0	0	0	1	1	0	0	0	0	1	2	2

The intermediate output of the differential evolution algorithm is shown in table 7.4.

Table 7.4: Successful simulation optimisation run of the flight-to-gate assignment problem of the validation case study.

Generation	f-count	Best f(x)	Mean population	Max constraint
1	8	133426207.49	58473402776	4
2	16	74881960.55	561612665.5	4
3	24	20424907.56	320811454.7	2
4	32	20424907.56	228912672.7	2
5	40	20424907.56	178144095.5	2
6	48	20424907.56	178144095.5	2
7	56	20424907.56	178144095.5	2
8	64	20424907.56	178144095.5	2
9	72	20424907.56	178144095.5	2
10	80	20424907.56	178144095.5	2
11	88	20424907.56	127390018.8	1
12	96	20424907.56	84361643.85	1
13	104	20424907.56	84361643.85	1
14	112	20424907.56	84361643.85	1
15	120	20424907.56	84361643.85	1
16	128	20424907.56	84361643.85	1
17	136	20424907.56	84361643.85	1
18	144	20424907.56	84361643.85	1
19	152	1976.22	66384965.33	0
20	160	1976.22	66384965.33	0
21	168	344.79	441326862.2	0
22	176	344.79	441326862.2	0
23	184	344.79	441326862.2	0
24	192	344.79	441326862.2	0

What can be seen is that the optimum was found after 21 generations. A total of 192 different solutions were evaluated using the agent-based model and simulation. The total amount of time needed to find this solution was a around seven days. Again, this is the major drawback of this method: computational-time. A total of six optimisations were needed to find the optimum presented here. Three of these runs converged and the other three did not.

The optimum objective function value which was found $\mathbf{f = 344.79\ s}$, was obtained with the following decision variable:

$$x_{opt} = [1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 2\ 3].$$

This means that the first four flights are assigned to gate C and the last four flights to gate D. Security checkpoint C and D have two and three X-ray scanners active, respectively. The security checkpoint queue times per time interval are shown in fig. 7.1.

Observed is that the security queue times for the different checkpoints are fairly close together. Only the final part (around interval number 70), the queue times for SC C are around one minute higher than the security queue times observed at SC D.

What the algorithm has done is assigning the flights with the expected highest number of passengers to gate D, and in addition assign three X-ray scanners to security checkpoint D. This meets the expectations, since the objective is to minimise the differences between queue times of the two checkpoints,

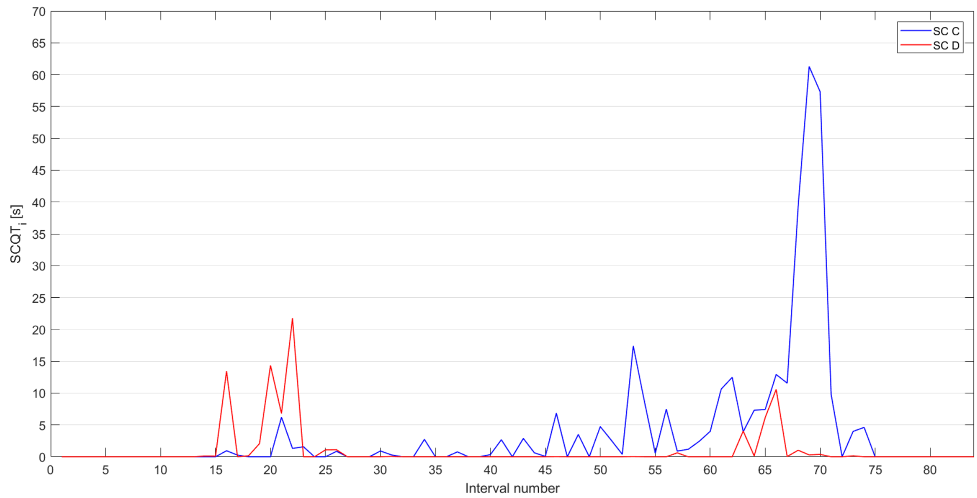


Figure 7.1: Security checkpoint queue times results of the flight-to-gate assignment by simulation optimisation.

then logically the highest number of operators will be assigned to the 'busiest' security checkpoint.

But why then is SC D used for these larger aircraft and not SC C? This is due to the minor difference in performance of the two security checkpoints. Assigning these aircraft in this way resulted in smaller differences in queue times, compared to the inverse of the above assignment. Furthermore, what can be observed is the large peak of high queue times round interval 68 at SC C. The reason for is peak is possibly due to the fact that this period of elevated queue times was unavoidable. The optimisation algorithm can only change the discrete events of assigning flights to gates, apparently the proposed assignment resulted in the smallest difference between the queue times experienced at SC C and D.

The optimal assignment found by the simulation optimisation method is assumed to be the global optimum. It has been tested by making manual changes to the assignment. However, none of the changes resulted in a lower objective function value. Furthermore, this result is assumed to be the actual result of the optimisation problem since it includes the actual agent-based model and simulation in the loop. Therefore, this result will be used as benchmark to compare the other methods against.

7.1.2. Regression model based optimisation

The second method that was used to solve the flight-to-gate assignment problem is the regression model based optimisation. This method is significantly faster than the simulation optimisation method (a couple seconds compared to days). Therefore, this method would be way easier to use for example on a weekly basis by airport managers.

The optimal decision variable found by the regression model based optimisation is:

$$x_{opt} = [1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 2\ 3].$$

Clearly, this assignment is different from the assignment found by the simulation optimisation algorithm. However, the assignment of the X-ray scanners are the same as was done by the simulation optimisation method. The objective function value (calculated by the regression meta-models) found for the above assignment was equal to $\mathbf{f = 5468.54\ s}$. This is way higher than the optimal value found by simulation optimisation. In general, the difference between the realisation and forecast is due to the lack in forecasting performance. Hence, the forecast produced by the regression models do not accurately estimate the queue times that would be observed from a simulation experiment.

The forecast of the regression models made of the optimal gate and X-ray scanner assignment is shown in fig. 7.2. In addition, the optimal assignment has been simulated in AATOM, which is also shown in fig. 7.2.

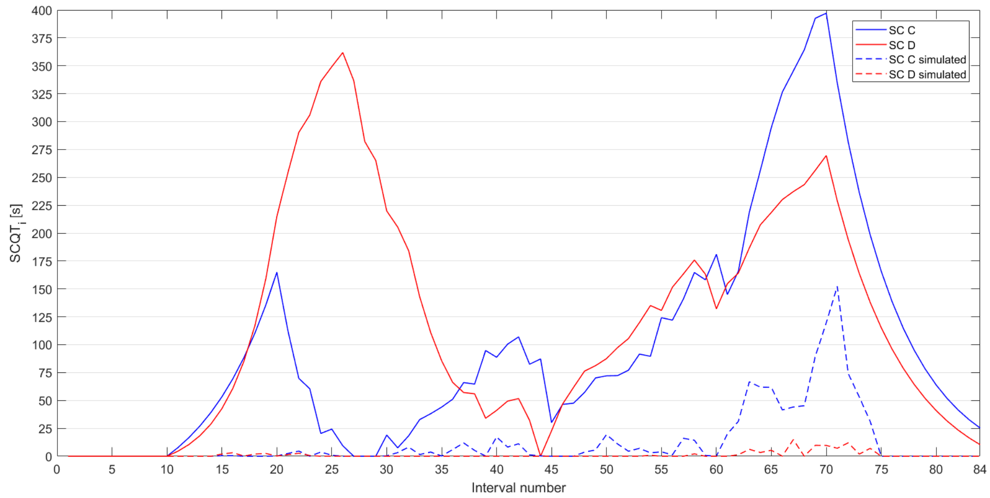


Figure 7.2: Validating security checkpoint queue times as a result of the flight-to-gate assignment by regression model based optimisation and the simulated queue times.

What can be seen is that the queue time forecast produced by the regression meta-models are fairly close to each other in the second half of the planning horizon (from interval 38 onwards). However, before interval 38 the queue times forecast for SC C are significantly lower than for SC D. Hence, the algorithm has done its job in finding an assignment that minimised the difference between queue times at SC C and SC D.

However, simulating the optimal gate and X-ray scanner assignment showed that the forecasts do not represent the agent-based model accurately (see fig. 7.2). The simulated queue times at SC D are for all intervals close to zero. Whereas, the regression meta-models forecast queue times up to six minutes. Similar results are found for SC C, but one could argue that the forecast's shape of the final peak of SC C is close to the peak's shape in the simulation. However, it is expected that the shape similarities is merely a coincidence since the evaluation of the forecast in section 6.3 showed no shape similarities in that area (see fig. 6.10). The objective function value would have been $f = 996.70\ s$ (actual objective function value), if it would have been calculated based on the simulation realisation of the optimal solution (found by the regression based optimisation).

What can be concluded is that the regression model based optimisation is not able to find the same optimum as the simulation optimisation method. Furthermore, the forecasts produced by the regression meta-model do not only estimate higher queue times (in this case), but shape-wise they are also completely off. Hence, the regression meta-models are not able to fully replace the agent-based model since they are not able to find the same optimum as the simulation optimisation method. The optimum found by the regression based optimisation however, is a close to the real optimal solution.

7.1.3. Gaussian radial basis function based optimisation

Finally, the Gaussian radial basis function based optimisation is tested on the above described flight-to-gate assignment problem. Also this method is significantly faster than the simulation optimisation method. However, it is to be seen whether or not this method will be able to find the same optimum as the simulation optimisation method and if the queue time' forecasts are similar to the agent-based simulation observed queue times.

The optimal decision variable found by the GRBF based optimisation is:

$$x_{opt} = [1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 2\ 3].$$

This is again a different optimum for the same flight-to-gate assignment problem. Clearly it failed to locate the true global optimum, found by the simulation optimisation method. The found optimal objective function value (calculated by the GRBF meta-models) is equal to $\mathbf{f = 1087.65\ s}$. Which is lower than the objective function value found by the regression based optimisation, but higher than the simulation optimisation objective function value. However, again two X-ray scanners were assigned to SC C and three X-ray scanners to SC D.

The GRBF models' forecast of the optimal aircraft and X-ray scanner assignment is shown in fig. 7.3. In addition, the found optimal assignment has been simulated in AATOM, which is also shown in fig. 7.3.

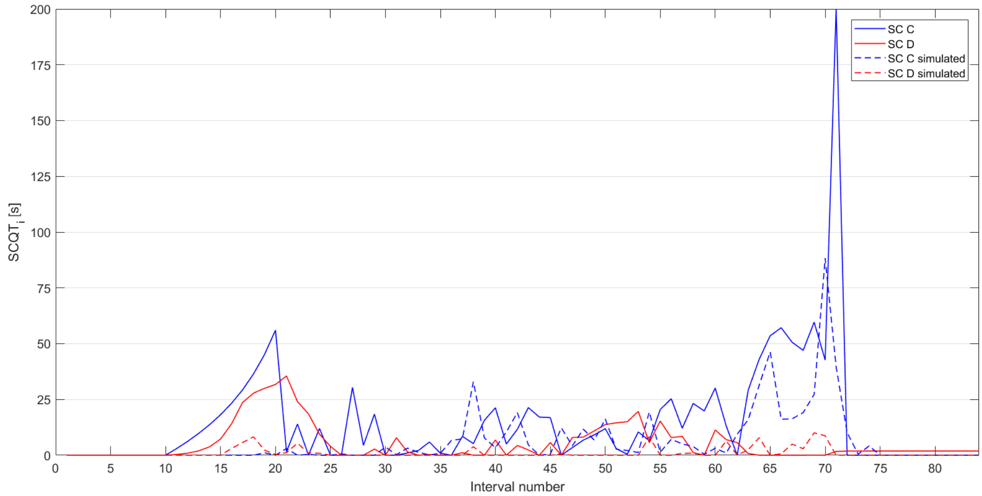


Figure 7.3: Validating security checkpoint queue times as a result of the flight-to-gate assignment by GRBF based optimisation and the simulated queue times.

What should be observed (again) is that the lines for the security checkpoint queue times (forecast) are close to each other at all time intervals. Compared to the regression based optimisation, the level of the forecast matches the level of the simulation output data better. This was also observed when examining the forecast performance of both models (figs. 6.15 and 6.16).

The forecast's shape for SC C seems to match the shape of the simulated data, especially around the last peak (from interval 62 onwards). Furthermore, in the middle section of the horizon, the GRBF meta-model for SC C seems to accurately tell when the queue times are going up or down. In the beginning of the planning horizon, the forecast shows a bad performance in estimating the appropriate level of queue time. This is probably due to the fact that the start-up phase (phase I) is represented by a bad performing model, since it is only dependent on the lagged security checkpoint queue time.

Little can be said about the forecast made for SC D, except for the fact that the level of queue time is close to the simulated queue time level.

Concluding, it can be said that the GRBF based optimisation is not able to locate the global optimum found by simulation optimisation. However, the objective function value would have been $f = 505.22\ s$, if it would have been calculated based on the simulation realisation of the optimal solution (found by the GRBF based optimisation). Hence, the method is able to locate a near optimal solution. In addition, the GRBF meta-models are able to predict at least the level of the security checkpoint queue time. Nonetheless, it should be concluded that the GRBF based optimisation is not able to fully replace the simulation optimisation strategy.

7.1.4. Summary validation case study

The results of the validation case study are summarised in table 7.5. The optimal objective function values found by the optimisation models are shown in the second column of the table and in the third column the objective function values calculated from the realisations of the optimal assignment are shown.

Table 7.5: Summary of validation case study showing the objective function value calculated from the model forecasts and from the realisation of the optimal assignments.

Optimisation model	f_{model} [s]	$f_{realisation}$ [s]
Simulation optimisation	344.79	344.79
Regression model	5468.54	996.70
GRBF model	1087.65	505.22

As discussed the simulation optimisation methodology is the only method that arrives at the true optimal assignment (with $f = 344.79s$). The optimal assignment found by the regression model based optimisation was the least optimal assignment of all three models, since the $f_{realisation}$ was the highest. Furthermore, the objective function value calculated by the regression models ($f = 5468.54s$) is higher than found by the GRBF meta-models. This could have been partially caused by the fact that the regression meta-models have the tendency to over-react (identified when fitting) to slight input changes. Which makes it difficult for the regression based optimisation to match the queue times experienced at the different security checkpoints. In terms of $f_{realisation}$, both the regression and GRBF meta-model based integrations are able to locate acceptable assignments relatively close to optimal. However, this could be a coincidence since the fitting and generalisation errors of both meta-model types were generally high.

7.2. Verification case study

After assessing the validity of the meta-model based integrations, the verification case study is performed to give the reader an idea of how the meta-model based optimisations would work once they are fully capable to replace the agent-based model and simulation. The main point of this case study is to see whether the proposed methodologies are able to locate the global/local minimum of a larger gate and resource assignment problem. As was seen in previous case study, and also during the fitting of the meta-models, the meta-models are currently not able to fully capture the dynamics of the agent-based model. The generalisation errors of the models were high and the forecast produced by the models was not consistent with the simulation results. During the current case study, one should forget about these limitations for the moment.

The current case study is constructed using the observations presented in table 4.2, in which the mix of aircraft observed at AAS was given. Furthermore, in the schedule used, it was tried to include the waves of arriving Schengen aircraft that are connected to departing non-Schengen aircraft. The schedule contained 44 flights that depart within 15 hours and 25 minutes. The main reason for this case study is to verify that the proposed optimisation method works the way it is designed. Hence, to see whether the following objective function is minimised:

$$\min f = \sum_{int=1}^{T} \frac{1}{m-1} \sum_{j=1}^{m} (SCQT_{int_j} - \overline{SCQT_{int}})^2,$$

where m are the security checkpoints A to D and T (total number of time intervals) was equal to $\frac{15\times60+25}{5} = 185$. The schedule developed for the current case study is presented in table 7.6. The fourth and the ninth column show whether the aircraft needs to be assigned to a Schengen (S) or non-Schengen (N-S) gate.

The DE settings used to find the optimum have been shown in table D.1 (appendix D), which are reported for the people that would like to replicate the current study. These settings have been found by

Table 7.6: Schedule used to test the regression model based optimisation and GRBF model based optimisation.

Flight	Arr time	Dep time	To	Size [pax]	Flight	Arr time	Dep time	To	Size [pax]
1	1:40	3:00	N-S	450	23	8:40	10:00	N-S	350
2	1:40	3:00	N-S	280	24	9:30	10:10	S	200
3	2:20	3:00	S	190	25	9:35	10:15	S	90
4	2:20	3:00	S	90	26	9:00	10:20	N-S	220
5	3:00	3:40	N-S	110	27	10:15	10:55	S	90
6	3:00	3:40	S	70	28	10:20	11:00	S	160
7	3:00	3:40	S	190	29	10:25	11:05	N-S	120
8	3:10	3:50	N-S	200	30	10:30	11:10	N-S	200
9	3:50	4:30	S	190	31	11:05	11:45	S	160
10	4:00	4:40	S	120	32	11:10	11:50	N-S	185
11	3:55	5:15	N-S	220	33	11:10	11:50	N-S	110
12	4:15	5:35	N-S	350	34	11:15	11:55	S	90
13	5:20	6:00	S	150	35	11:55	12:35	S	120
14	5:45	6:25	S	100	36	11:55	12:35	S	185
15	5:35	6:55	N-S	410	37	12:00	13:20	N-S	220
16	5:45	7:05	N-S	210	38	12:20	13:40	N-S	350
17	6:50	7:30	S	90	39	13:30	14:10	S	120
18	7:05	7:45	S	200	40	13:45	14:25	S	190
19	7:20	8:00	N-S	100	41	13:45	15:05	N-S	220
20	7:40	8:20	N-S	180	42	13:50	15:10	N-S	450
21	8:10	8:50	S	120	43	14:40	15:20	S	90
22	8:50	9:30	S	195	44	14:45	15:25	S	150

performing the optimisations in abundance. These settings give the highest probability of convergence towards the potential global optimum.

7.2.1. Regression model based optimisation

First, the flight-to-gate assignment problem was solved using the regression model based optimisation. The optimisation algorithm has been run several times, in order to make sure that the optimum found had the highest probability of being a global optimum. In section 7.2.4 the performance of the algorithm will be elaborated upon. Each of the optimisation runs took around 25 minutes to solve. This was due to the high number of decision variables (180) and the high population size which was required (100). In addition, the flatness of the objective function will have contributed as well.

The gate and X-ray scanner assignment that resulted in the smallest difference in waiting times amongst the checkpoints was found, and is shown in table 7.7.

The security checkpoint queue times for the optimal assignment is shown in fig. 7.4. The algorithm finally assigned three X-ray scanners to checkpoint A and C and two X-ray scanners to B and D. The assignment resulted in a optimal objective function value of $\mathbf{f = 1.601E{+}06\ s^2}$.

It took around 250 generations to find the optimal assignment, meaning that 12500 individuals were evaluated. What can be seen from the figure is that the queue times at SC B are quite close to the queue times at SC D. Also, the queue times experienced at SC A are close to zero for the entire plan-

Table 7.7: Optimal gate assignment found by the regression model based optimisation for the schedule presented in table 7.6.

Gate A - 3 X-ray scanners			Gate B - 2 X-ray scanners		
Flight no	Dep time	Size [pax]	Flight no	Dep time	Size [pax]
1	3:00	450	2	3:00	280
5	3:40	110	8	3:50	200
11	5:15	220	12	5:35	350
15	6:55	410	16	7:05	210
19	8:00	100	20	8:20	180
23	10:00	350	26	10:20	220
29	11:05	120	30	11:10	200
32	11:50	185	33	11:50	110
37	13:20	220	38	13:40	350
41	15:05	220	42	15:10	450

Gate C - 3 X-ray scanners			Gate D - 2 X-ray scanners		
Flight no	Dep time	Size [pax]	Flight no	Dep time	Size [pax]
3	3:00	190	4	3:00	90
6	3:40	70	7	3:40	190
9	4:30	190	10	4:40	120
13	6:00	150	14	6:25	100
17	7:30	90	18	7:45	200
21	8:50	120	22	9:30	195
24	10:10	200	25	10:15	90
27	10:55	90	28	11:00	160
31	11:45	160	34	11:55	90
35	12:35	120	36	12:35	185
39	14:10	120	40	14:25	190
43	15:20	90	44	15:25	150

ning horizon, except for the beginning of the planning. This major disadvantage is due to the fact that the model in phase I is only dependent on the lagged observed queue time, which was already explained when fitting the regression meta-models. Except for phase I, the queue times observed at SC A are very close to the queue times at SC C.

Overall, the queue times seem to be fairly close to each other. Hence, this indicates that the method is able to find the optimal assignment, leading to the most equal spread of queue times amongst the security checkpoints. However, due to the fact that the assignment is a discrete event assignment, the security checkpoints queue times will in most cases not be able to exactly match each other. If one would want to achieve a perfect match of queue times amongst stations, then the passenger streams should be decoupled. This means that the path restriction assumption is being dropped, giving the airport manager the possibility to guide any sub-group of passengers to any security point. Decoupling the passenger processes (e.g. check-in facilities with security checkpoints) would introduce new challenges which will need to be solved, but also decrease the causal relationship between the arrival distributions and the queue build-up at a specific station.

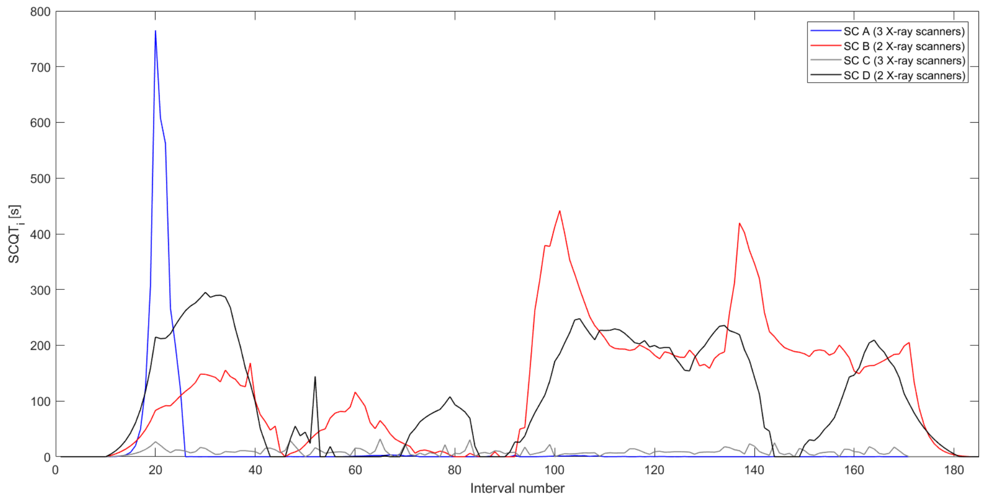


Figure 7.4: Security queue times as a result of the optimal assignment found by the regression model based optimisation. The objective function value for this assignment is equal to $f = 1.601\text{E+}06\ s^2$.

7.2.2. Gaussian radial basis function based optimisation

The same flight-to-gate assignment problem was solved by making use of the Gaussian radial basis function based optimisation. For this optimisation method, the same settings as in the regression based optimisation were used for to find the optimum.

The smallest differences in queue times amongst the checkpoints was obtained by the gate assignment shown in table 7.8.

What can be seen is that the assignment is completely different from the assignment found by the regression model based optimisation method. However, both methods assign three X-ray scanners to SC A and C, and two X-ray scanners to gates B and D. Due to the complex nature of the model, it is difficult to say why certain decisions have been made by the optimisation tool.

What these models have in common is that the meta-models for 3 X-ray scanners generally predict queue times very close to zero for schedules with medium sized aircraft in them. Therefore, the optimisation will look for a schedule sequence/mix, such that the queue times of the checkpoints that have two X-ray scanners active are as close to zero as possible, as was seen in fig. 7.4. Apparently, security checkpoint B and D perform better than A and C with two X-ray scanners for this schedule. Note that in the first case study, this was exactly the opposite.

The predicted queue times for the checkpoints are shown in fig. 7.5.

What can be seen is that the queue times at SC A to C are fairly close to each other (below 100s per time interval). However, the model is not able to suppress the queue times that will be apparent at SC D, during two peak periods. If it were possible, the algorithm would probably have liked to activate another X-ray scanner at SC D. Note, that the objective function value $\mathbf{f = 3.236E{+}06\ s^2}$ is larger than the objective function value from the regression based optimisation. This is probably due to the high peaks from SC D, because the other periods show superior performance w.r.t. the queue times in fig. 7.4. In addition, the forecasts produced by GRBF is by definition different from the regression forecast.

To show the difference, the optimal assignment found by the GRBF based optimisation can be used by the regression meta-models to forecast the security checkpoint queue times. In this way one could again see the difference between the two meta-models. This has been shown in fig. 7.6.

One aspect that immediately stands out, is the fact that the queue time peaks of SC D are still there.

Table 7.8: Optimal gate assignment found by the GRBF model based optimisation for the schedule presented in table 7.6.

Gate A - 3 X-ray scanners			Gate B - 2 X-ray scanners		
Flight no	Dep time	Size [pax]	Flight no	Dep time	Size [pax]
1	3:00	450	2	3:00	280
8	3:50	200	5	3:40	110
12	5:35	350	11	5:15	220
15	6:55	410	16	7:05	210
19	8:00	100	20	8:20	180
23	10:00	350	26	10:20	220
29	11:05	120	30	11:10	200
32	11:50	185	33	11:50	110
38	13:40	350	37	13:20	220
41	15:05	220	42	15:10	450

Gate C - 3 X-ray scanners			Gate D - 2 X-ray scanners		
Flight no	Dep time	Size [pax]	Flight no	Dep time	Size [pax]
4	3:00	90	3	3:00	190
6	3:40	70	7	3:40	190
9	4:30	190	10	4:40	120
14	6:25	100	13	6:00	150
18	7:45	200	17	7:30	90
22	9:30	195	21	8:50	120
24	10:10	200	25	10:15	90
27	10:55	90	28	11:00	160
34	11:55	90	31	11:45	160
35	12:35	120	36	12:35	185
40	14:25	190	39	14:10	120
43	15:20	90	44	15:25	150

However, the tip of these peaks are around 300 seconds lower based on the regression meta-models. Furthermore, the regression meta-models for SC B have a stronger reaction to increases in expected passengers (reduction in inter arrival time). Therefore, the peaks for security checkpoint B are (as expected) higher for the regression meta-models than the GRBF meta-model's forecast. The queue times observed at security checkpoint A and C are very close to zero (except for phase I of SC A). However, the queue times experienced at SC B and D are farther apart and therefore overall the queue times are less spread amongst checkpoints.

The objective function value ($2.0263\text{E+}06\ s^2$) calculated by the regression meta-models of the GRBF model optimal assignment is logically higher than the optimum found by the regression based optimisation model ($f = 1.601\text{E+}06\ s^2$). But, it is lower than found by the GRBF based optimisation model.

7.2.3. Summary verification case study

The results of the verification case study are summarised in table 7.9. What should be observed from the table is the fact that both meta-model based optimisations resulted in different optimal solutions. In

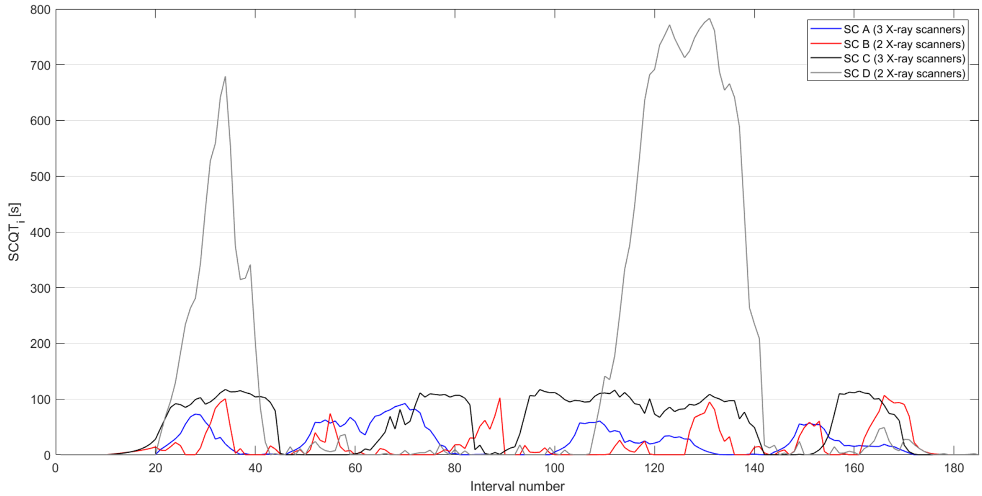


Figure 7.5: Security queue times as a result of the optimal assignment found by the GRBF model based optimisation. The objective function value for this assignment is equal to $f = 3.236\text{E+06}\ s^2$.

Table 7.9: Summary of verification case study showing the objective function value calculated by the regression and GRBF meta-models from the optimal assignment found by the regression based and GRBF based optimisation.

	Objective function value calculated by	
Assignment found by	Regression meta-models [s^2]	GRBF meta-models [s^2]
Regression based optimisation	1.601E+06	4.176E+06
GRBF based optimisation	2.026E+06	3.236E+06

addition, the table indicates that both optimisation models could be able to locate optimal solutions. As for example the optimal solution found by the regression based optimisation had an objective function value of 1.061E+06s^2 and the alternative solution (result of GRBF based optimisation) resulted in an objective function value of 2.026E+06s^2. Hence, the solution found by the regression based optimisation is better than the alternative. The same holds for the GRBF based optimisation. Furthermore, manual trials of potential solutions also did not result in improvements of the proposed optimal solutions. Hence, this verified that the meta-model based optimisations were able to locate (global) optimal solutions.

7.2.4. Differential evolution algorithm performance

To conclude this chapter, a final word should be said about the performance of the optimisation algorithm, since the optimisation strategies that were developed (figs. 5.4 and 6.17) did not converge every time to the global optimum or even to a feasible solution.

The performance of the DE algorithm is dependent on selecting the right parameter settings for the right characteristics of the considered problem. In addition, the DE algorithm has difficulties in handling integer linear problems with equality constraints, and the convexity of the solution space also plays a major role.

Selecting the parameter settings that result in the highest probability of converging runs was, together with understanding the algorithm performance, a matter of trial and error. It was known that the objective function value was dependent on the number of intervals considered. A longer planning horizon required higher penalty values. In addition, the infeasible solutions could have very low objective function values. Consider for example the case that no flights are assigned, then the objective function value ($f = 0\ s^2$) since there are no queues apparent at the airport. This makes it difficult for the DE to move away from the infeasible solutions, unless the penalty values are set to high values. The level of

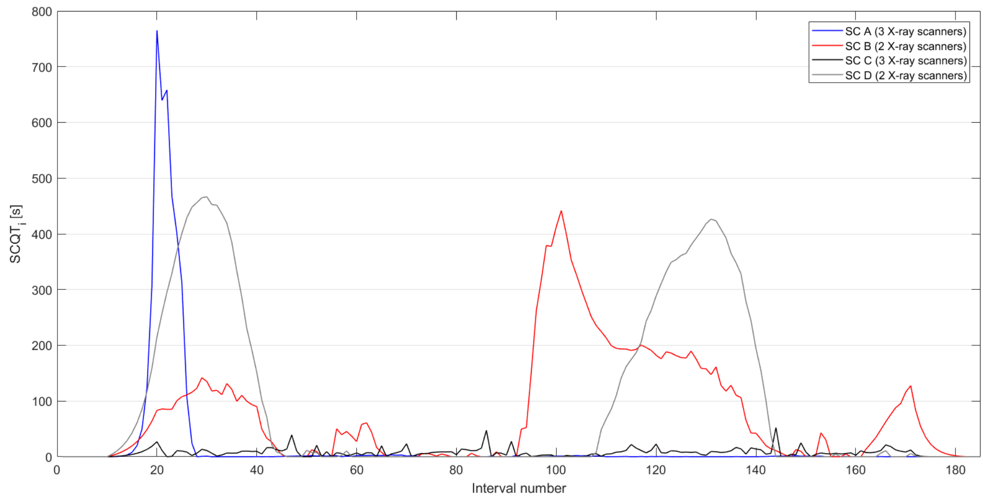


Figure 7.6: Security queue times predicted by the regression meta-models. The optimal assignment found by the GRBF model based optimisation. The objective function value for this assignment is equal to $f = 2.0263\text{E+}06\ s^2$.

these penalty values were obtained by performing many trial and error runs.

Secondly, the DE had difficulties with handling equality constraints. This was a major concern since the equality constraints in the posed flight-to-gate assignment problems were very strict. As the majority of the equality constraints ('$= 1$') only involved two decision variables and the decision variables were binary. This only leaves two possibilities, assigning the flight to one gate or to the other. Hence, the solution space is quite narrow, which is difficult for a DE algorithm to handle (since it could easily 'jump' over the feasible solution space).

The focus of the current study was only on four gates (two Schengen and two non-Schengen) of a fictitious airport. Future research could easily extend the proposed problem by including more gates, which could relax the majority of the strict constraints. Therefore, it is believed that the DE algorithm would perform even better in future research on a similar topic.

Thirdly, also the convexity of the solution space plays a role in the speed of convergence as well as the ability to locate the global minimum. A close to similar objective function value can be achieved by different solutions, since the objective function value is a summation of the variances per time interval. As this was the case, one of the remedies was to select a large population during each iteration. However, a large population size also means that the algorithm is less efficient, since it needs to evaluate a lot of feasible and infeasible solutions.

Especially during the simulation optimisation one wants to have the correct parameter settings such that convergence would occur, since each optimisation run took around seven days. This was done by doing trial and error runs of the regression model based optimisation (on the first case), before starting on the simulation optimisation. Nonetheless, the first couple simulation optimisation runs did not converge to feasible solutions or did not converge to the, what was believed to be the, global optimum.

All the solutions presented in the current research have been challenged by many optimisation runs as well as manual trial and error runs. It cannot be guaranteed that all the optima found are global optima, however the optima belong definitely to the best of the local optima since trial and error did not result in improved solutions.

8

Discussion and implications

In this chapter a synthesis of all intermediary results, which were given throughout the report, will be given. Meetings at Rotterdam The Hague Airport have been an important input for this chapter. In these meetings the scientific contribution of the thesis was presented and knowledge was gained on the flight-to-gate assignment practice. First, a discussion of the results and implications of the assumptions are given in section 8.1. Secondly, in section 8.2 the implications of the results are given, by making a distinction between research implications and industry implications.

8.1. Discussion of results

In this section the results presented throughout the report will be discussed. This is done by discussing the overall picture of the results in the light of the research objective. First the integration success will be discussed. Secondly, the quality of the agent-based abstraction models will be elaborated on and finally the individual quality of the meta-models will be discussed.

Since the research objective is to integrate an agent-based model with an flight-to-gate assignment it is necessary to discuss the quality of the integration. In this thesis, the focus has been on the security checkpoint queues as it was found that the security checkpoint is the main landside source of airside delay. Two novel methodologies have been developed to integrate an agent-based model into a flight-to-gate assignment optimisation. The scientific contribution is substantial, since up till now the integration of an agent-based model with a flight-to-gate assignment optimisation has not yet been investigated nor performed.

The first method presented was a direct integration. The objective function of the flight-to-gate assignment was the realisation of the agent-based model, which was dependent on the gate and resource assignment. The direct integration method proposed was based on the simulation optimisation framework, which includes a simulation model in an optimisation strategy. The system was solved using a differential evolution algorithm. During the initial case study, both advantages and disadvantages of a direct integration were recognised.

The main advantage of this method is the fact that there is little loss in detail due to approximation, compared to the indirect methods. The only loss was induced by the time interval assumption and the number of replication runs to estimate the queue time per interval. The time interval assumption induced loss due to the fact that the queue time experienced within the five minute time interval was not constant. The second is that the number of replications to estimate the queue times during the intervals also create a loss in accuracy. Ideally, the number of replication runs would be so large that extra runs would not change the variance of the average queue times per interval any more. In general, ten runs were done to estimate a single scenario, except for cases where the spread of the queue time observed per scenario (at a time interval) was relatively large. The limited number replication runs will also have induced loss in detail. In the simulation optimisation method, it was decided to estimate the average queueing time per time interval from 10 simulation runs of the same scenario. However, the actual average queue time during the time interval would better approximated by e.g. 100 replication simulation runs. Therefore, there is loss induced by only performing 10 replication simulation runs.

The main disadvantage of this method is the computational time. As discussed, it took around one week to complete one fairly simple optimisation run, without guaranteed success. The problem solved in the initial case only involved two gates and eight flights. Therefore, it is believed that solving a larger case with more flights and gates would lead to even more unacceptable computational times. Therefore, this way of integration is believed to be inappropriate for an airport manager to use on a weekly basis.

The second method of integration made use of agent-based model abstractions. This method was proposed to overcome the main disadvantage of the direct integration method. The main advantage of this indirect integration method is the fact that, compared to the direct integration, the optimisation would only take a fraction of the time needed by the direct integration method.

This decrease in computational time however, was achieved by sacrificing the amount of details that were present in the agent-based model. In addition to the loss of detail that was present in the direct integration method, additional loss is induced by the approximation errors. During the fitting stage of the meta-modelling process, the goal was to minimise the approximation errors. It was observed that the fitting errors of the meta-models (both regression and GRBF) were generally not acceptable. The point estimate errors of models created for phase II (in-operation phase), were up to 2 minutes and 50 seconds. The severity of such an error is dependent on the scenario. In cases where the queue times at the security checkpoints reach up to forty minutes, these errors are actually quite good (93% accuracy). However, in scenarios where low queue times are observed these errors are not acceptable. The significant differences in queue times observed is due to three factors. The first cause is the diversity of aircraft sizes that is assumed to depart from the fictitious airport. These range between small aircraft such as the Embraer 190 and large aircraft such as the Airbus A380. Assigning two X-ray scanners to a security checkpoint that only handles the E190s, could result in no queue build up. Whereas the two X-ray scanners would not be able to avoid queue build up if assigned to a security checkpoint that only handles A380s. Secondly, the frequency of the flights also strongly affects the differences in queue times observed between scenarios. Finally, the sensitivity of the additional X-ray scanner plays another significant role. The settings used for the X-ray scanners created situations in which one could reduce the observed queue times from 40 minutes to only a couple of seconds by opening an extra X-ray scanner. Further calibration of the model is therefore necessary, since these situations will not occur in reality.

In addition to the fitting and generalisation errors of the meta-models, which indicates the meta-model's quality of abstraction, forecasts were created with both meta-model types. Furthermore, a case study was conducted to test the validity of optimal assignment found by the meta-model based optimisation.

The meta-model variables included the lagged estimated queue time ($SCQT_{i-1}$). This is, compared to the other variables (inter-arrival times), not a strictly exogenous variable and therefore the point estimate errors are not the only performance measures of interest. Forecasts were produced by both meta-models of an arbitrary gate assignment and compared to the agent-based model and simulation outcome of the same gate assignment. Mixed results were observed.

The GRBF meta-models of security checkpoint B proved to be able to approximate the actual queue behaviour. It was expected that the meta-models for SC A and B, and C and D had to show similar behaviour as the pairs experience the same type of passenger streams. The difference between the GRBF meta-models produced for SC C and SC D (2 X-ray scanners) was remarkable. The GRBF meta-models for gate D produced a large over-forecast, whereas the meta-models for SC C performed well. The difference between GRBF meta-models for the presented schedule of SC A and B was more acceptable.

The regression meta-models for SC A and B proved to both over-forecast the queue times, but were fairly close to each other. In general, the regression meta-models were very eager to predict growths in queue times, compared to GRBF meta-models. However, the regression models were less eager to predict reductions in queue time. This behaviour was a result of the form of the model and resulted many times in structural over-forecasts in phase II.

Furthermore, the predicted queue time behaviour (shape of the graph) did not consistently follow the observed queue time behaviour from the simulation. However, it was observed that overall the GRBF meta-models were able to predict the level of the queue times better than the regression meta-models.

Global sensitivity analysis was conducted to assess the importance of the parameters in four meta-models ($SCQT^{C,2,2}$, $SCQT^{B,2,2}$ for both the regression and GRBF meta-model). In the non-parametric models, the $SCQT_{i-1}$ variable was most important. However, in the regression (parametric) models the $SCQT_{i-1}$ variable was less important. The importance ranking of the lagged inter-arrival time variables were dependent on both the specific security checkpoint and the meta-model type.

Before drawing major conclusions, the results of the indirect integrations were compared to the result of the direct integration. It was assumed that the simulation optimisation (direct integration) would result in the real global optimal gate assignment. Therefore, if the meta-model based optimisations would be able to locate the same optimum, then there would be evidence that the meta-models are able to replace the agent-based model in the optimisation strategy. However, both meta-model optimisations resulted in different 'optimal' assignments. Hence, the validity of the developed indirect integration optimisation method depends on the quality of the meta-models.

The main result of this study has been the development of two methodologies to integrate an agent-based model and simulation with a flight-to-gate assignment optimisation. The scientific gap that was existent has been closed with a decent research, that encourages further research in this area. Furthermore, this study describes in detail all the steps that have been taken or explored to make the integration possible. These include: understanding the agent-based model and the emergent behaviour, developing necessary agent-based model features, agent-based model calibration, feature selection, design of experiments, data generation, meta-model fitting and evaluation, integration with optimisation and optimisation evaluation. One of the main challenges when integrating was to match the level of the agent-based model (micro/passenger level) with the flight-to-gate assignment optimisation (macro/airport level), without losing too much detail of the agent-based model and simulation.

8.1.1. Implications of the assumptions

To be able to perform research within a limited amount of time and scope, assumptions needed to be made. Many of the assumption made in this thesis affected the results of the research. It is important to assess the implications of the main assumptions.

One of the main assumption that ties the two paradigms (ABMS and OR) together is the path restriction assumption. Passengers heading for a specific gate, take a specific path corresponding to that gate. This assumption was necessary since parametric models require a clear correlation between the independent variables and the dependent variables. If this assumption was not made, people would be able to choose a security checkpoint themselves in a random way. Then there would be little correlation between the arrival rates of passengers at the entrance of the airport for a specific flight and the queue times observed at one of the security checkpoints. If the gate assignment was completely uncorrelated with the path passengers take towards the gate, then the regression meta-models could even have performed worse. Hence, this assumption is critical to the regression meta-model integration.

Secondly, the proposed integration is also heavily dependent on the passenger knowledge assumption, which assumes knowledge about the amount of passengers departing per aircraft and a deterministic arrival distribution. In reality, these arrival distributions might be non-deterministic, depending on the destination and size of the airport. Furthermore, there could be factors influencing these arrival distributions that are unknown. The proposed methodology is heavily dependent on the direct relationship between assigning aircraft and the known arrival of passengers. In situations where the arrival distributions are uncertain, the proposed methodology might not be valid.

Furthermore, the passenger knowledge assumption also assumes that the number of passengers departing per flight is known. In reality the airport has no knowledge in advance about the exact number of passenger departing per flight, as became clear during a meeting with Duty Manager Operations Dennis Gerharts of Rotterdam The Hague Airport. The knowledge they do have is the aircraft type and the time it is supposed to arrive and depart. The current research aims to precisely predict queue times per security checkpoint based on the number of passenger arriving per flight. If in a case instead of 200 passengers only 90 passengers arrive for a certain flight, this could result in a different gate and resource allocation. Hence, the optimum found by the integrated model is only valid if the passenger knowledge assumption holds.

In addition, it is assumed that the agent-based model is fully calibrated. This assumption ensures that if an optimal gate and resource allocation is found, then it is also in reality an optimal allocation.

Finally, the input variables chosen for the meta-models and the type of the meta-models largely determine the quality of the meta-models. It was assumed that the two X-ray scanner data sets determined the input-variables taken for the meta-models and for the GRBF meta-models also the spread (σ). Ideally, the meta-modelling process steps had to be repeated for the three X-ray scanner data sets, potentially leading to different variables and settings. Hence, the assumption could have resulted in less performing three X-ray scanner meta-models. Furthermore, the time interval standard assumption in combination with the meta-model types, affected the selection of variables. Hence, assuming a different time interval standard would have resulted in a different variable selection and hence, different meta-model performances.

8.2. Implications of the results

In this section the implications of the results will be elaborated on. This is done by first assessing the implications for future research and secondly the implications for the industry.

8.2.1. Research implications

It is important to consider the implications for future research since the aim of current study is to open a new area of research. This research was part of a cluster at the Delft University of Technology, in which multiple students worked on different topics within the subject of agent-based modelling and airport terminal operations, under supervision of dr. Alexei Sharpanskykh and Stef Janssen. The current study was one of the first studies within the cluster that used the entire agent-based model. Hence, an important contribution of the current research has been the identification and resolution of issues and improvements found during the research.

Before starting the research it was assumed that finding an appropriate meta-model, that would be able to capture the dynamics prevalent in an agent-based model, would be difficult. This is due to the fact that meta-models are ‘simple’ equations with limited flexibility, whereas an agent-based model takes a bottom up approach which aims at representing reality. This was the main reason during current research to create different models for the phases and number of X-ray scanners active. However, even within these restricted usage areas, the meta-models still have little explanatory power.

It can be argued whether or not a set of meta-models will ever be able to capture and represent processes in an agent-based model. The response of many agent-based model processes are highly non-linear which makes it very difficult to determine variables that are highly correlated with the observed response. It is assumed that only a carefully managed set of probably many meta-models will be able to approximate the response of an agent-based model and simulation. However, if this set of variables and relations can be determined is to be seen. At least, with the proposed parameter selection, meta-model types and fitting methods used in the current study, did not result in meta-models with satisfactory performance.

The main contribution to the existing literature are the developed of methodologies to integrate the agent-based modelling and simulation paradigm with the operations research paradigm. To the author’s knowledge, little to no integrations have been made between the two paradigms. The current study has opened up the research area by providing a solid initial attempt by providing methodologies to integrate the paradigms in a direct and indirect way. Especially, the area of simulation optimisation has proven to be a very promising way to integrate an agent-based model for airport passengers with an flight-to-gate assignment optimisation.

The contribution to the flight-to-gate assignment literature is limited. This is mainly because the current state of the flight-to-gate assignment literature is already in an advance stage. Therefore, the contribution of this research is mainly on the introduction of dynamic passenger models in the flight-to-gate assignment problems. In addition, the research provides a solid static model that can be extended to a robust model with more constraints which are representative for the airside considerations.

8.2.2. Industry implications

Next, it is important to consider the implications for the industry. In specific for the airport managers responsible for the airside and landside operations.

During the meetings at RTHA it became clear that the methods current research proposes it not the current practice, in terms of flight-to-gate assignments. Flight-to-gate assignments at RTHA do not involve very complex mathematical tools whereas at AAS many more considerations are taken into account, mainly financial considerations.

It was acknowledged that there is a clear disconnect between the airside and landside management. The current research tries to find a precise relationship between assignments of flights-to-gates and the minimisation of passenger delays in the terminal, for an airport with only four gates. One could question why this study contributes to the industry practice. However, this study could be seen as a way for airports to steer passenger flows at the airport. For example at Schiphol airport it could be interesting to manage the passenger flows, not between gates but, between piers. Such that the utilisation of the pier facilities are equally spread. The result could be increased retail revenues and improved passenger experience. However, the proposed method would only be valid if the agent-based model if fully calibrated and validated.

Currently, airports are managing passenger flows in a reactive manner. Cameras and sophisticated tracking software are installed to monitor the terminal processes and the prevalence of congestions. The proposed methodology tries to manages the passenger flows in a pro-active manner. Expected passenger flows are predicted and the congestions (at the security checkpoints) minimised or avoided. The proposed method would however require lots of data on the arrival distributions, passenger specific characteristics, decision trade-offs etc. A proper implementation of the proposed method at an airport would result in less real-time passenger related adjustments.

A comment should be made about the consequence of the proposed method at a real airport. The method proposed, optimises the spread of queue times by assigning gates and resources. Hence it assumes that the manager in charge of gate assignments, will be the same as or closely involved with the security manager. At RTHA this is not the case. As an alternative, the proposed method could also be used when the one responsible of the gate assignment knows how many resources are employed during a day and where the resources are located.

Currently, the model is not yet ready to be used in practice. This would require calibration of the model, more research into the meta-modelling of an agent-based model, and speeding up the computation in case of the simulation optimisation method. However, in the future, the model could ease the work of airport managers (landside and airside), improve passenger experience at airports and increase retail revenue by steering passenger streams over the airport.

9 Conclusion and recommendations

This chapter will present the most important conclusions and recommendations. The conclusion will be discussed in section 9.1, where the research questions asked in chapter 3 will be answered. In section 9.2, recommendations for future research will be given.

9.1. Conclusion

This research has taken the first steps towards developing a methodology to integrate an agent-based model for airport passengers with a flight-to-gate assignment optimisation. This section will give the main conclusions based on the research questions presented in chapter 3.

Q1: What airport areas are important to consider when integrating a multi agent-based model for passenger processes in a flight-to-gate assignment problem?

In this research, the main focus has been on the security checkpoint, or more specifically, the queues of security checkpoints. This focus was set based on the found delay sources at airports in literature. However, this does not mean that the security checkpoint is the only area that had to be considered.

Passengers enter the airport at different times before departure. Some passengers need to pass by the check-in desks to check-in and/or to drop-off luggage. Furthermore, passengers might dwell around the airport before heading towards the security checkpoint. Including these areas increases the detail of the model, and makes it closer to reality. Passengers do not all walk the same distances, as was taken into account by many of the older static flight-to-gate assignment problems. Therefore, if one wants to make the passenger experience better in terms of queue time at the security checkpoint, then the agent-based model should represent realistic airport terminal processes. This can be achieved by considering all areas and processes, before the security checkpoint (e.g. entrance area, check-in area, goodbye area), when integrating the agent-based model with a gate and resource assignment optimisation.

Q2: What variables used in the agent-based model from the areas above, could be used in the flight-to-gate assignment model?

In this research it was chosen to restrict the paths passengers could take towards their designated gate. The result of this restriction was that there were actually variables of the agent-based model that could be used in the flight-to-gate assignment. In reality, the gate assignment is only weakly correlated with the paths passengers take, since the gate only defines a rough path that passengers will take towards their gate. Hence, the flight-to-gate decision variables decide the rough path for passengers to follow to their flights.

As the objective function of the optimisation problem was to spread the queue times as equally among security checkpoints as possible, the security queue time was the main output variable of interest. The queue time was affected by the number of passengers departing per flight, and thus the assignment of those flights to gates.

Furthermore, in the meta-modelling based integration method, the inter-arrival times of passengers were also taken as the input variables for the meta-models. The inter-arrival times were the average time between arrival of passengers at the entrance area. The number of passengers arriving per five minute time interval was determined by making the arrival distribution assumption. This assumed that the number of passengers departing per flight and their arrival distribution were known in advance. Based on the behaviour of the data, different combination of these variables (including the lagged security queue time) were made.

In addition, the number of X-ray scanners open per security checkpoint were taken into account. These variables were taken as decision variables in the optimisation problem. Since these directly affected the queue time behaviour per security checkpoint.

Other important variables were the number of operators per check-in counter, passenger transfer percentage, business/leisure ratio and other parameters shown in table A.1. These variables were not directly included in the assignment problem, but defined the experiment setting.

Q3: Can the agent-based model be integrated in a direct way or by creating meta-models that are able to capture the dynamic relations?

In this research, the agent-based model was integrated with a flight-to-gate assignment model in both a direct and indirect way.

The direct integration was done by the simulation optimisation method. Since only little detail was lost by averaging queue times over five minute intervals, the optimisation of direct integration was assumed to lead to the real global optimal assignment. However, this method had a major disadvantage, namely, the computational time required to find the optimal assignment. The differential evolution algorithm used to find the optimal assignment examined both the feasible and non-feasible solution space. Especially the latter space sometimes required a computation time of over 2 hours for a single simulation run. Furthermore, the flatness of the objective function also contributed to the long optimisation runs, without guaranteed success. The complete optimisation took in the order of days. This did not guaranteed a feasible solution nor a global optimum. Hence, this method would not be useful to use on a daily basis in real operations.

Therefore the indirect method of integration was developed. The agent-based model was replaced by sets of abstractions, or meta-models. Two types of meta-models were developed: Regression meta-models and Gaussian radial basis function meta-models. The models were created per phase (I, II, III) , per security checkpoint queue (A, B, C, D) and per number of X-ray scanners active (2 or 3). The models have been fitted on the same fitting data set. In phase I and III the only significant variable was the lagged dependent variable. This indicates that the other variables had no significant effect on the security queue time in phase I and phase III. Such an autoregressive model is known to have bad forecasting performance. Hence, in these phases the meta-models are not able to capture the dynamic behaviour of the governing agent-based model.

It was shown that both types of models under-performed (in terms of fitting error) in phase II with two X-ray scanners active. The error was in general around two minutes, which is large knowing that the queue time can be as low as a couple seconds in this phase and setting. However, the queue times in this phase could also be around 40 minutes, in which case the fitting errors are satisfactory.

In addition, the generalisation errors were assessed by testing the models on a validation data set. In most meta-models the generalisation errors were higher than the fitting errors. In addition, the generalisation errors were leading in the selection process of the GRBF meta-models.

The meta-model sets were also tested in terms of forecasting performance. Mixed results were observed. In general, the regression meta-models were shown to be less able to predict drastic decreases in queue time, leading in general to an over-forecasts in phase II. The GRBF meta-models also showed mixed results. However, the drastic increases or decreases in queue time were better approached by the GRBF meta-models. In some cases the GRBF models were able to approximate the shape of real security checkpoint queue times quite accurately. However, the results found were not consistent and therefore it cannot be said that the constructed meta-model sets are able to capture all the dynamic relations present in the agent-based model.

The indirect integration, meta-models with the flight-to-gate assignment optimisation, could only be achieved by assuming what was mentioned under **Q2**. The result was a computationally fast converging algorithm compared to the simulation optimisation method. It was verified, by means of a verification case study, that the indirect integrated methods resulted were optimising for equal spread of queue times among security checkpoint queues. However, the optimisation algorithm suffered from the objective function flatness, which made it difficult for the algorithm to locate the global optimum.

Q4: Is the integrated model credible, does it reflect the agent-based model and reality?

Especially the latter part of this question is difficult to answer as this requires data or expert knowledge of the real underlying system. In addition to the fitting error, generalisation error and forecast performance, the credibility of the meta-modelling based integration was tested using a validation case study.

As was explained, the optimum found by the simulation optimisation method was assumed to be the actual optimal gate and resource assignment. Hence, assessing the difference between the optima found, for the same departure schedule, by the meta-model based optimisation methods and the simulation optimisation method would give insight in the validity of the meta-model based optimisation method.

What was observed was that all three methods gave different gate and resource assignments as optimal assignment. Hence, the meta-models are not a valid replacement of the agent-based model in the simulation optimisation strategy. However, it was found that both meta-model based optimisations (Regression and GRBF) were able to locate a close to optimal assignment to the posed flight-to-gate and resource allocation problem.

Finally, the validity of the meta-models with respect to reality was difficult to assess. Global sensitivity analysis was conducted to determine the most important parameters per model. However, expert knowledge is needed to assess whether these parameters are the most valuable per situation in reality. In general, the predicted security checkpoint queue time (in phase II) went up when the number of passengers arriving per time interval increased and went down when the number of passenger arriving per time interval decreased. Furthermore, increasing the number of X-ray scanners (increasing the service rate) resulted in lower experienced queue times. This is in line with common knowledge about queue behaviour.

To conclude, it is possible to integrate an agent-based model with an flight-to-gate assignment by imposing the mentioned assumptions. The direct integration of an agent-based model into a optimisation algorithm results in a very slow optimisation. However, the optimum found is the actual optimum. Integrating the agent-based model by means of meta-models has been proven to be more difficult. Especially the emergent behaviour in the agent-based model output data are difficult to approximate by meta-models. The meta-models are not yet a valid replacement for the agent-based model in the optimisation strategy. Once the agent-based model is fully calibrated, the meta-models are fully validated and have good performances, airport managers could use the integrated tool to guide passenger flows over the airport such that congestions are avoided or minimised.

9.2. Recommendations for future research

In this section the recommendations for future research will be given. In general, the results found and shown in this research are dependent on all assumptions made and variables considered. Hence, making different assumptions and/or changing the variables could lead to better or worse performing models.

Before conducting any further research using the AATOM simulator, it is recommended to calibrate the model. For example, it was found that the security checkpoint queue time behaviour was very sensitive to opening a new X-ray scanner. However, this could be due to the fact that the agent-based model is not yet fully calibrated with real airport data. Therefore, to be able to draw conclusions about whether found optima with e.g. a simulation optimisation method are real, one could research the calibration of the AATOM model. This can drastically change the relations studied in this research.

The major disadvantage of the simulation optimisation method, was the fact that it took days to find a solution to a relatively simple problem. Hence, future research could aim to increase the computational speed of the method by changing the agent-based model or by using a different solution algorithm.

Furthermore, the meta-model types that were used in current research are only two of the many possibilities. Therefore, future research could focus on developing new meta-models or improve the meta-models by using different parameters (including higher order terms). Meta-modelling of an agent-based model could in itself be a very interesting research topic. In this way, different variables can be included, which do not have to be related to the flight-to-gate assignment problem.

In the current study the path restriction was imposed, which defined the path corresponding to a gate. Future research could try to decouple the flows from e.g. one specific check-in desk to the other specific security checkpoint. This way, passenger flows could even be directed in between processes, and not only by assigning flights to gates.

Additional research could be done in the creation of the fitting data sets (design of experiments). In the current study it is tried to evenly spread the design data points across the solution space. However, since the solution space is of high-dimensionality, it could be done in different ways. Therefore, a similar study could be conducted by changing the design of experiments or to optimise the design of experiments. This could improve the performance of the meta-models.

Another interesting study that could be conducted is to examine the real-time reassignment problems that airport managers encounter. A coordinating agent could be put in place, that can reassign flights-to-gates when congestions occur in the terminal or flights are delayed.

As discussed, the data split proposed in the current research could have been done differently. The current study split the data into 24 distinct data sets (per phase, per SC and per number of X-ray scanners). This resulted in meta-models with high fitting and generalisation errors. Future research could split the data differently to improve the meta-models' performances.

Finally, in the current research the variables (regression modelling) and spread (GRBF modelling) used in the meta-models of the 3 X-ray scanner data sets were determined by the 2 X-ray scanner data sets. Ideally, the fitting and model selection procedure would be done for each data set. Future research could replicate the current but select variables and spread for every separate data set.

Bibliography

Airports Council International. Best Practice Guidelines: Airport Service Level Agreement Framework. Technical report, ACI, 2014.

Airports Council International. Annual Report 2016. Technical report, 2016.

Aktel, A., Yagmahan, B., Özcan, T., Yenisey, M. M., and Sansarci, E. The comparison of the meta-heuristic algorithms performances on airport gate assignment problem. *Transportation Research Procedia*, 22(2016):469–478, 2017. ISSN 23521465. doi: 10.1016/j.trpro.2017.03.061. URL http://dx.doi.org/10.1016/j.trpro.2017.03.061.

Aronson, J., Liang, T., and Turban, E. Decision support systems and intelligent systems, 2005. URL http://sutlib2.sut.ac.th/sut{_}contents/H86360.pdf.

Babić, O., Teodorović, D., and Tošić, V. Aircraft Stand Assignment to Minimize Walking. *Journal of Transportation Engineering*, 110(1):55–66, 1984. ISSN 0733-947X. doi: 10.1061/(ASCE)0733-947X(1984)110:1(55). URL http://ascelibrary.org/doi/10.1061/{%}28ASCE{%}290733-947X{%}281984{%}29110{%}3A1{%}2855{%}29.

Barton, R. R. Metamodels for simulation input-output relations. *Winter Simulation Conference*, 9(x): 289–299, 1992. doi: 10.1145/167293.167352.

Belsley, D. A., Kuh, E., and Welsch, R. E. *Regression diagnostics: Identifying influential data and sources of collinearity*. 2005.

Biles, W. E. A gradient-regression search procedure for simulation experimentation. *Conference on Winter simulation*, pages 491–497, 1974. ISSN 0717-6163. doi: 10.1007/s13398-014-0173-7.2.

Bolat, A. Procedures for providing robust gate assignments for arriving aircrafts. *European Journal of Operational Research*, 120(1):63–80, 2000. ISSN 03772217. doi: 10.1016/S0377-2217(98)00375-0.

Bolat, A. A. Models and a genetic algorithm for static aircraft-gate assignment problem. 52(10):1107–1120, 2001. doi: 10.1198/016214502388618870.

Bonabeau, E. Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl. 3):7280–7287, 2002. ISSN 0027-8424. doi: 10.1073/pnas.082080899. URL http://www.ncbi.nlm.nih.gov/pubmed/12011407.

Bouras, A., Ghaleb, M. A., Suryahatmaja, U. S., and Salem, A. M. The airport gate assignment problem: A survey. *Scientific World Journal*, 2014, 2014. ISSN 1537744X. doi: 10.1155/2014/923859.

Brazile, R. P. and Swigger, K. M. Gates, an airline gate assignment and tracking expert system, 1988. ISSN 0885-9000.

Burmeister, B., Haddadi, a., and Matylis, G. Application of multi-agent systems in traffic and transportation. *IEE Proceedings - Software Engineering*, 144(1):51–60, 1997. ISSN 1364-5080. doi: 10.1049/ip-sen:19971023. URL http://digital-library.theiet.org/content/journals/10.1049/ip-sen{_}19971023.

Carson, Y. and Maria, A. Simulation Optimization: Methods and Applications. *Proceedings of the 29th conference on Winter simulation (1997)*, pages 118–126, 1997. ISSN 0743-1902. doi: 10.1145/268437.268460.

Casado, A. S., Laguna, M., and Pacheco, J. Heuristical labour scheduling to optimize airport passenger flows. *Journal of the Operational Research Society*, 56(6):649–658, 2004. doi: 10.1057/palgrave.jors.2601859.

Castaing, J., Mukherjee, I., Cohn, A., Hurwitz, L., Nguyen, A., and Müller, J. J. Reducing airport gate blockage in passenger aviation: Models and analysis. *Computers and Operations Research*, 65: 189–199, 2016. ISSN 03050548. doi: 10.1016/j.cor.2014.02.011. URL http://dx.doi.org/10.1016/j.cor.2014.02.011.

Chang, C. Flight sequencing and gate assignment at airport hubs. *Transportation Research*, pages 53–54, 1996.

Charnes, J. M. Statistical Analysis of Output Processes. *Proceedings of the 1993 Winter Simulation Conference*, pages 41–49, 1993.

Chen, B. A Review of the Applications of Agent Technology in Traffic and Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):485–497, 2010. ISSN 1524-9050. doi: 10.1109/TITS.2010.2048313.

Cheng, C. H., Ho, S. C., and Kwan, C. L. The use of meta-heuristics for airport gate assignment. *Elsevier*, 39(16):12430–12437, 2012. ISSN 09574174. doi: 10.1016/j.eswa.2012.04.071. URL http://dx.doi.org/10.1016/j.eswa.2012.04.071.

Cheng, L., Reddy, V., Fookes, C., and Yarlagadda, P. K. D. V. Impact of passenger group dynamics on an airport evacuation process using an agent-based model. *Proceedings - 2014 International Conference on Computational Science and Computational Intelligence, CSCI 2014*, 2:161–167, 2014. doi: 10.1109/CSCI.2014.111.

de Neufville, R. and Odoni, A. *Airport systems - Planning, design, and management*. 2013.

Diepen, G., Van Den Akker, J. M., Hoogeveen, J. A., and Smeltink, J. W. Finding a robust assignment of flights to gates at Amsterdam Airport Schiphol. *Journal of Scheduling*, 15(6):703–715, 2012. ISSN 10946136. doi: 10.1007/s10951-012-0292-y.

Ding, H., Lim, A., Rodrigues, B., and Zhu, Y. New heuristics for over-constrained flight to gate assignments. pages 760–768, 2004. doi: 10.1057/palgrave.jors.2601736.

Dorndorf, U., Drexl, A., Nikulin, Y., and Pesch, E. Flight gate scheduling: State-of-the-art and recent developments. *Omega*, 35(3):326–334, 2005. ISSN 03050483. doi: 10.1016/j.omega.2005.07.001.

Drexl, A. and Nikulin, Y. Multicriteria airport gate assignment and Pareto simulated annealing. *Iie Transactions*, 40(4):385–397, 2008. ISSN 0740-817X. doi: Doi10.1080/07408170701416673.

Eilon, S. and Mathewson, S. A Simulation Study for the Design of an Air Terminal Building. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(4):308–317, 1973. ISSN 21682909. doi: 10.1109/TSMC.1973.4309241.

Şeker, M. and Noyan, N. Stochastic optimization models for the airport gate assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(2):438–459, 2012. ISSN 13665545. doi: 10.1016/j.tre.2011.10.008.

Eurocontrol. Network Operations Report 2017. Technical report, 2017.

Fu, M. C., Price, C. C., Zhu, J., and Hillier, F. S. *Handbook of Simulation Optimization Associate Series Editor*. 2015. ISBN 9781493913831. doi: 10.1007/978-1-4939-1384-8_9.

Garg, A. and Tai, K. Comparison of Regression Analysis , Artificial Neural Network and Genetic Programming in Handling the Multicolliearity Problem. pages 24–29, 2012.

Gatersleben, M. R. and van der Weij, S. W. ANALYSIS AND SIMULATION OF PASSENGER FLOWS IN AN AIRPORT TERMINAL. pages 1226–1231, 1999.

Gosling, G. D. Design of an expert system for aircraft gate assignment. *Transportation Research Part A: General*, 24(1):59–69, 1990. ISSN 01912607. doi: 10.1016/0191-2607(90)90071-D.

Gu, Y. and Chung, C. A. Genetic Algorithm Approach to Aircraft Gate Reassignment Problem. *Journal of Transportation Engineering*, 125(5):384–389, 1999. ISSN 0733-947X@e1943-5436. doi: 10.1061/(ASCE)0733-947X(1999)125:5(384). URL http://cedb.asce.org/cgi/WWWdisplay.cgi?118730.

Haghani, A. and Chen, M.-C. Optimizing Gate Assignments at Airport Terminals. *Elsevier*, 1997. ISSN 0041-1655. doi: 10.1287/trsc.19.2.173.

Hamzawi, S. G. Management and planning of airport gate capacity: a microcomputer□based gate assignment simulation model. *Transportation Planning and Technology*, 11(3):189–202, 1986. ISSN 0308-1060. doi: 10.1080/03081068608717341. URL http://www.tandfonline.com/doi/abs/10.1080/03081068608717341.

Hartigan, J. A. and Wong, M. A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society*, 28(1):100–108, 1979. URL http://www.jstor.org/stable/2346830.

Holland, J. H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.*, volume 8. 1975. ISBN 0262581116. doi: 10.1137/1018105. URL http://www.citeulike.org/group/664/article/400721.

Huang, D., Allen, T. T., Notz, W. I., and Zeng, N. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, 34(3):441–466, 2006. ISSN 09255001. doi: 10.1007/s10898-005-2454-3.

IATA and ACI. Airport Development Reference Manual (ADRM). Technical report, 2014.

Janssen, S. and Sharpanskykh, A. Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection. 10349:132–143, 2017. doi: 10.1007/978-3-319-59930-4. URL http://link.springer.com/10.1007/978-3-319-59930-4.

Janssen, S., Blok, A. N., and Knol, A. AATOM - An Agent-based Airport Terminal Operations Model. 2017.

Jin, R., Chen, W., and Simpson, T. W. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23(1):1–13, 2001. ISSN 1615147X. doi: 10.1007/s00158-001-0160-4.

Kingman, J. F. C. General Stochastic Processes in the Theory of Queueus. *Stochastics and Sotchastics Reports*, 43(3-4):221–260, 1963.

Kleijen, J. P. C. Validation of models: statistical techniques and data availability. *Simulation Conference Proceedings, 1999 Winter*, 1:647–654, 1999.

Kleijnen, J. P. C. *Design and analysis of simulation experiments*, volume volume 230. 2008. ISBN 978-3-319-18086-1. doi: 10.1007/978-3-319-18087-8.

Kleijnen, J. P. C. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, 2009. ISSN 03772217. doi: 10.1016/j.ejor.2007.10.013.

Li, C. Airport gate assignment A hybrid model and implementation. *arXiv preprint arXiv:0903.2528*, 2009. URL http://arxiv.org/abs/0903.2528.

Lim, A. and Wang, F. Robust Airport Gate Assignment. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2005:74–81, 2005. ISSN 10823409. doi: 10.1109/ICTAI.2005.110.

Ma, W. *Agent-based model of passenger flows in airport terminals*. PhD thesis, 2013.

Manfren, M., Aste, N., and Moshksar, R. Calibration and uncertainty analysis for computer models – A meta-model based approach for integrated building energy simulation. *Appl. Energy*, 103:627–641, 2013.

Mangoubi, R. S. and Mathaisel, D. F. X. Optimizing Gate Assignments at Airport Terminals. *Transportation Science*, 19(2):173–188, 1985. ISSN 0041-1655. doi: 10.1287/trsc.19.2.173.

Maule, A. J. and Svenson, O. *Time pressure and stress in human judgment and decision making*. 1993. ISBN 978-1-4419-3233-4. doi: 10.1007/978-1-4757-6846-6.

Mehrotra, K., Mohan, C. K., and Ranka, S. *Elements of artificial neural networks*. 1997. ISBN 0262133288.

Miyoung, S. and Goel, A. Empirical data modeling in software engineering using radial basis functions. *IEEE Transactions on Software Engineering*, 26(6):567–576, 2000. ISSN 00985589. doi: 10.1109/32.852743. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=852743.

Miyoung Shin, Sargent, R., and Goel, A. Gaussian radial basis functions for simulation metamodeling. *Proceedings of the Winter Simulation Conference*, 1(October):483–488, 2014. doi: 10.1109/WSC.2002.1172920. URL http://ieeexplore.ieee.org/document/1172920/.

O'Sullivan, D. and Haklay, M. Agent-based models and individualism: Is the world agent-based? *Environment and Planning A*, 32(8):1409–1425, 2000. ISSN 0308518X. doi: 10.1068/a32140.

Pan, Q.-Q., Tasgetiren, M. F., and Liang, Y.-C. A discrete differential evolution algorithm for the permutation flowshop scheduling problem. *Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO '07*, page 126, 2007. doi: 10.1145/1276958.1276976. URL http://portal.acm.org/citation.cfm?doid=1276958.1276976.

Pintea, C. M., Pop, P. C., Chira, C., and Dumitrescu, D. A hybrid ant-based system for gate assignment problem. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5271 LNAI:273–280, 2008. ISSN 03029743. doi: 10.1007/978-3-540-87656-4_34.

Royal Schiphol Group Mediarelaties. Traffic and transport figures for April 2018, 2018. URL https://news.schiphol.com/traffic-and-transport-figures-for-april-2018/.

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., Edwards, D. D., and Jonathan, S. J. S. *A Modern Approach*, volume 13. 1995. ISBN 0137903952. doi: 10.1007/s11894-010-0163-7. URL http://www.ncbi.nlm.nih.gov/pubmed/21104209.

Saltelli, A., Tarantola, S., Campolongo, F., and Ratto, M. *Sensitivity Analysis in Practice*. ISBN 0470870931.

Schiphol Group. Royal Schiphol Group Annual Report 2016. 2016.

Schultz, M. and Fricke, H. Managing passenger handling at airport terminals individual-based approach for modeling the stochastic passenger behavior. *Proceedings of the 9th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2011*, pages 438–447, 2011. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84941777942{&}partnerID=40{&}md5=0d8883984bad3eef540123bb8dae3fc8.

Šeda, M., Šedová, J., and Horký, M. Models and Simulations of Queueing Systems. *International Conference on Soft Computing-MENDEL.*, 2016.

Šeda, M., Ošmera, P., and Petrucha, J. Systems with Queueing and their Simulation. 5(1):865–871, 2011. ISSN 2010376X.

Shin, M. and Goel, A. Radial basis function model development and analysis using the SG algorithm. Technical report, Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, New York 13244., 1998.

Shin, M. *Design and evaluation of radial basis function model for function approximation*. 1998.

Shin, M. and Park, C. A Radial Basis Function Approach to Pattern Recognition and Its Applications. *ETRI Journal*, 22(2):1–10, 2000. ISSN 1225-6463. doi: 10.4218/etrij.00.0100.0201. URL http://doi.wiley.com/10.4218/etrij.00.0100.0201.

Snee, R. D. Validation of Regression Models: Methods and Examples. *Technometrics*, 19(4):415–428, 1977.

Spans, J. An integrated approach for efficient and resilient airport management An integrated approach for efficient and resilient airport management: Literature review. Technical report, 2017.

Storn, R. and Price, K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, pages 341–359, 1997.

Su, Y. Y. and Srihari, K. A knowledge based aircraft-gate assignment advisor. *Computers and Industrial Engineering*, 25(1-4):123–126, 1993. ISSN 03608352. doi: 10.1016/0360-8352(93)90236-Q.

Teknomo, K. and Gerilla, G. Sensitivity analysis and validation of multi-agents pedestrian model. *Journal of the Eastern Asia Society for Transportation Studies*, 6(1995):198–213, 2005. URL http://www.easts.info/on-line/journal{_}06/198.pdf.

van Beers, W. C. M. and Kleijnen, J. P. C. Kriging for interpolation in random simulation. *Journal of the Operational Research Society*, 54(3):255–262, 2003. ISSN 0160-5682. doi: 10.1057/palgrave.jors.2601492.

van Rhee, C. Gate assignment using linear programming. Technical report, The Hague, 1992.

van Schaijk, O. R. and Visser, H. G. Robust Flight-to-Gate Assignment Using Flight Presence Probabilities. 2017.

Vemuganti, R. R. *Handbook of Combinatorial Optimization: Volume1–3*. 1999. ISBN 978-1-4613-0303-9. doi: 10.1007/978-1-4613-0303-9_9.

Weiss, G. *Multiagent Systems*. 2001.

Woolridge, J. M. *Introductory Econometrics*. 2012. ISBN 9781111531041.

Wray, B., Palmer, A., Bejou, D., and Wray, B. Using Neural Network Analysis to Evaluate Buyer-Seller Relationships. 1994.

Xu, J. and Bailey, G. The Airport Gate Assignment Problem: Mathematical model and a tabu search algorithm. *Proceedings of the Hawaii International Conference on System Sciences*, 00(c):77, 2001. ISSN 10603425. doi: 10.1109/HICSS.2001.926327.

Yan, S. and Huo, C. M. Optimization of multiple objective gate assignments. *Transportation Research Part A: Policy and Practice*, 35(5):413–432, 2001. ISSN 09658564. doi: 10.1016/S0965-8564(99)00065-8.

Yan, S. and Tang, C. H. A heuristic approach for airport gate assignments for stochastic flight delays. *European Journal of Operational Research*, 180(2):547–567, 2007. ISSN 03772217. doi: 10.1016/j.ejor.2006.05.002.

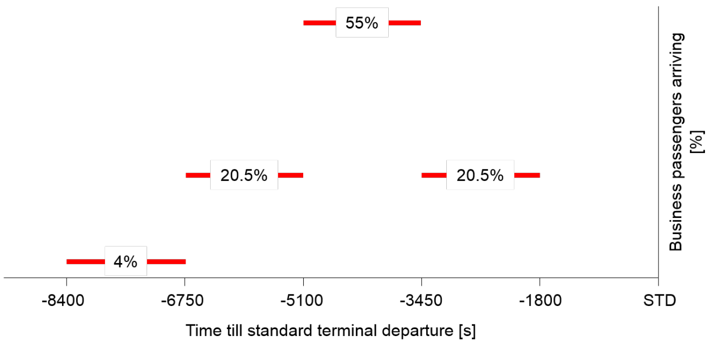
Yan, S., Shieh, C. Y., and Chen, M. A simulation framework for evaluating airport gate assignments. *Transportation Research Part A: Policy and Practice*, 36(10):885–898, 2002. ISSN 09658564. doi: 10.1016/S0965-8564(01)00045-3.

Zhang, Q. and Li, H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007. ISSN 1089-778X. doi: 10.1109/TEVC.2007.892759. URL http://ieeexplore.ieee.org/document/4358754/{%}0Ahttp://ieeexplore.ieee.org/ielx5/4235/4384509/04358754.pdf?tp={&}arnumber=4358754{&}isnumber=4384509.

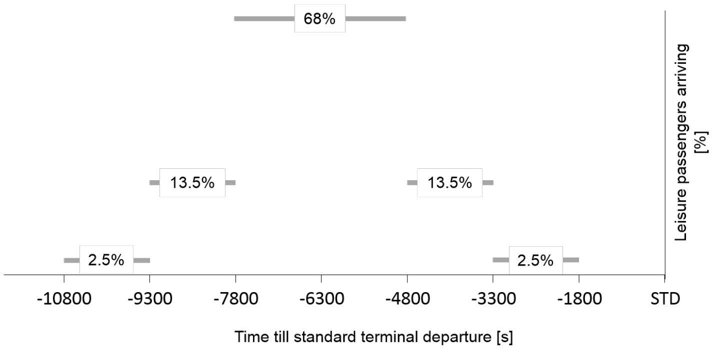
Zuur, A. F., Leno, E. N., and Smith, G. M. *Analysing Ecological Data*. 2007. ISBN 978-0-387-45967-7. doi: 10.1198/016214508000000715. URL https://www.tandfonline.com/doi/full/10.1198/016214508000000715.

A

Appendix



(a) Number of business passengers arriving per time interval as a percentage of the total number of business passengers.



(b) Number of leisure passengers arriving per time interval as a percentage of the total number of leisure passengers.

Figure A.1: Assumed passenger arrival distributions.

Table A.1: Overview of parameters implemented in AATOM.

Parameters	Standard value
Leisure Passenger	
V (m/s) ; N(μ, σ^2)	(1.19, 0.25)
P[Goodbye]	0.30
Goodbye time ; N(μ, σ^2)	(600,200)
Luggage complexity ; N(μ, σ^2)	(0.5, 0.15)
P[Carry-on]	0.50
P[Checked-in]	0.50
Business passenger	
Business percentage	0.32
V (m/s) ; N(μ, σ^2)	(1.38, 0.21)
P[Goodbye]	0.00
Luggage complexity ; N(μ, σ^2)	(0.8, 0.07)
P[Carry-on]	0.50
P[Checked-in]	0.50
Transfer passenger	
Transfer percentage	0.38
V (m/s) ; N(μ, σ^2)	(1.19, 0.25)
Check-in facility	
Service time ; N(μ, σ^2)	(60,6)
Security checkpoint	
Luggage check activity ; N(μ, σ^2)	(60,6)
Xray check ; N(μ, σ^2)	(60,6)
$t_{luggagedrop}$; N(μ, σ^2)	(54.6,36.09)
$t_{luggagecollect}$; N(μ, σ^2)	(71.5,54.95)
P[random check]	0
ETD check ; N(μ, σ^2)	(34.8, 15.2)
Border control point	
Service time ; N(μ, σ^2)	(30,6)

B
Appendix

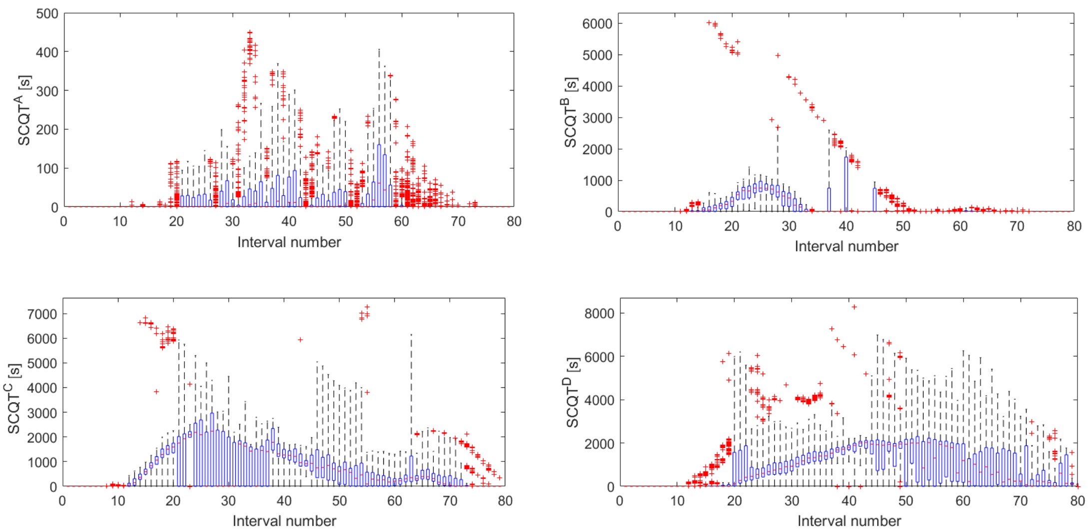


Figure B.1: A scenario in outliers need to be removed from the data set.

Table B.1: Belsey colliniarity diagnostics' variance decomposition. A condition index larger than 30 indicates colliniarity. Hence, there is no colliniarity in the selected variables.

sValue	condIdx	IAT_{i-1}	IAT_{i-5}	$SCQT_{i-1}$	$IAT_{i-5} \cdot SCQT_{i-1}$
1.581	1	0.0394	0.0333	0.0194	0.0198
1.0802	1.4637	0.0725	0.1346	0.0329	0.0313
0.4913	3.2182	0.879	0.8182	0.0003	0.0102
0.3035	5.2093	0.0092	0.0139	0.9474	0.9388

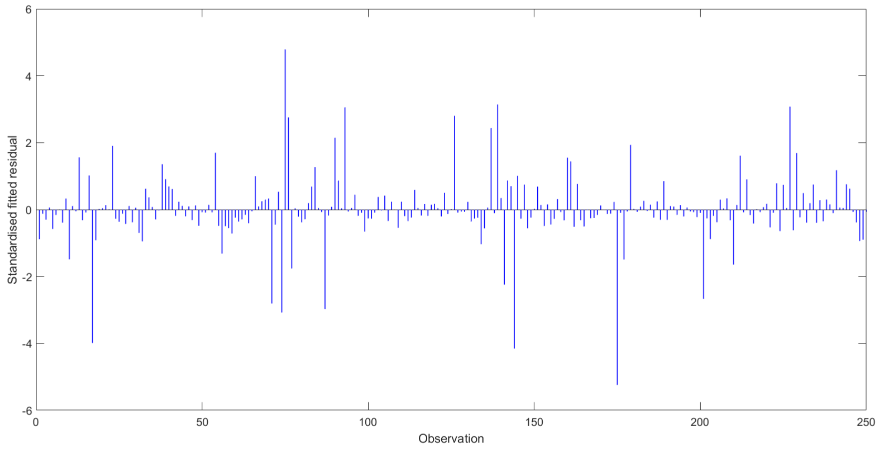


Figure B.2: Standardised residuals of the regression in table 6.6, showing a white noise process.

Table B.2: Regression meta-model for $SCQT_i^{C,2,3}$, R^2_{adj} = 0.881 RMSE = 31.2s.

	Estimate	SE	tStat	pValue
(Intercept)	-6.0525	8.0901	-0.74815	0.4551
IAT_{i-1}	0.15765	0.32408	0.48645	0.62709
IAT_{i-3}	0.9792	0.43343	2.2592	0.024766
IAT_{i-4}	-0.64577	0.31671	-2.039	0.042541
IAT_{i-5}	0.007052	0.081465	0.086567	0.93109
$SCQT_{i-1}$	1.2187	0.041911	29.079	8.97E-81
$IAT_{i-1}IAT_{i-3}$	-0.00492	0.006091	-0.80735	0.42026
$IAT_{i-4}SCQT_{i-1}$	-0.0046	0.003509	-1.3124	0.19065
$IAT_{i-5}SCQT_{i-1}$	-0.0187	0.004839	-3.8646	0.000143

Table B.3: Regression meta-model for$SCQT_i^{A,2,2}$, R^2_{adj} = 0.957 RMSE = 106 s.

	Estimate	SE	tStat	pValue
(Intercept)	68.145	27.685	2.4614	0.014544
IAT_i	-1.7354	0.93598	-1.8541	0.064956
IAT_{i-2}	-0.23057	0.8384	-0.27501	0.78354
IAT_{i-3}	0.73993	0.88554	0.83557	0.40423
IAT_{i-4}	-1.5916	0.96234	-1.6539	0.099456
$SCQT_{i-1}$	1.1278	0.034529	32.664	2.69E-90
IAT_iIAT_{i-4}	0.022577	0.011359	1.9876	0.047994
IAT_iSCQT_{i-1}	0.015737	0.003538	4.4485	1.32E-05
$IAT_{i-2}SCQT_{i-1}$	-0.01685	0.003963	-4.2511	3.05E-05
$IAT_{i-3}SCQT_{i-1}$	-0.01125	0.003585	-3.1387	0.001909

Table B.4: Regression meta-model for $SCQT_i^{A,2,3}$, R^2_{adj} = 0.956 RMSE = 8.65 s.

	Estimate	SE	tStat	pValue
(Intercept)	2.7004	2.1083	1.2808	0.20149
IAT_i	-0.0584	0.066903	-0.87288	0.3836
IAT_{i-2}	-0.00984	0.059501	-0.16545	0.86873
IAT_{i-3}	-0.0019	0.064464	-0.02943	0.97655
IAT_{i-4}	-0.02515	0.065917	-0.38151	0.70316
$SCQT_{i-1}$	1.7212	0.14925	11.532	9.00E-25
$IAT_i{:}IAT_{i-4}$	0.000718	0.000919	0.78075	0.43572
IAT_iSCQT_{i-1}	-0.04773	0.009723	-4.9085	1.70E-06
$IAT_{i-2}SCQT_{i-1}$	-0.05804	0.010625	-5.4627	1.17E-07
$IAT_{i-3}SCQT_{i-1}$	0.046447	0.009396	4.9432	1.44E-06

Table B.5: Regression meta-model for $SCQT_i^{B,2,2}$, R^2_{adj} = 0.956 RMSE = 82.7 s.

	Estimate	SE	tStat	pValue
(Intercept)	50.347	16.492	3.0529	0.0025193
IAT_i	-0.10217	0.29441	-0.34705	0.72886
IAT_{i-3}	0.90619	0.46934	1.9308	0.054677
IAT_{i-4}	-1.7874	0.62111	-2.8778	0.0043626
IAT_{i-5}	-1.1128	0.49674	-2.2401	0.025992
$SCQT_{i-1}$	0.89667	0.02189	40.963	7.97E-111
IAT_iSCQT_{i-1}	0.001595	0.000465	3.4315	0.00070576
$IAT_{i-4}IAT_{i-5}$	0.017767	0.005453	3.2584	0.0012812

Table B.6: Regression meta-model for $SCQT_i^{B,2,3}$, R^2_{adj} = 0.901 RMSE = 9.25 s.

	Estimate	SE	tStat	pValue
(Intercept)	2.3735	1.6536	1.4354	0.15247
IAT_i	0.018255	0.03419	0.53394	0.59387
IAT_{i-3}	-0.051	0.050887	-1.0023	0.31721
IAT_{i-4}	-0.02146	0.069283	-0.3097	0.75706
IAT_{i-5}	-0.05237	0.061484	-0.85174	0.3952
$SCQT_{i-1}$	0.86916	0.036301	23.943	9.10E-66
IAT_iSCQT_{i-1}	0.001873	0.000588	3.1849	0.001638
$IAT_{i-4}IAT_{i-5}$	0.001127	0.000563	2.0014	0.046466

Table B.7: Regression meta-model for $SCQT_i^{D,2,2}$, R^2_{adj} = 0.947 RMSE = 135 s.

	Estimate	SE	tStat	pValue
(Intercept)	142.67	46.505	3.0679	0.0024003
IAT_{i-1}	-2.6675	1.732	-1.5402	0.12482
IAT_{i-2}	-4.2563	2.1121	-2.0152	0.044983
IAT_{i-4}	0.51608	1.3259	0.38923	0.69745
$SCQT_{i-1}$	1.0383	0.048779	21.287	1.84E-57
$IAT_{i-1}IAT_{i-2}$	0.060026	0.029376	2.0434	0.042092
$IAT_{i-4}SCQT_{i-1}$	-0.00802	0.003281	-2.445	0.015194

Table B.8: Regression meta-model for $SCQT_i^{D,2,3}$, R^2_{adj} = 0.953 RMSE = 36.8 s.

	Estimate	SE	tStat	pValue
(Intercept)	6.5719	9.1188	0.72069	0.47179
IAT_{i-1}	-0.33729	0.43936	-0.76769	0.44342
IAT_{i-2}	0.018906	0.51337	0.036827	0.97065
IAT_{i-4}	-0.00648	0.35213	-0.01839	0.98534
$SCQT_{i-1}$	1.0739	0.044111	24.345	4.04E-67
$IAT_{i-1}IAT_{i-2}$	0.003353	0.00715	0.46889	0.63957
$IAT_{i-4}SCQT_{i-1}$	-0.00549	0.002829	-1.9406	0.05346

Table B.9: Regression meta-model for $SCQT_i^{A,1,2}$, R^2_{adj} = 0.99 RMSE = 9.39 s.

	Estimate	SE	tStat	pValue
(Intercept)	1.3802	0.95662	1.4428	0.15227
$SCQT_{i-1}$	1.3238	0.013543	97.75	1.69E-99

Table B.10: Regression meta-model for $SCQT_i^{A,1,3}$, R^2_{adj} = 0.828 RMSE = 2.9 s.

	Estimate	SE	tStat	pValue
(Intercept)	0.1246	0.29477	0.42271	0.67343
$SCQT_{i-1}$	2.4897	0.11402	21.834	2.00E-39

Table B.11: Regression meta-model for $SCQT_i^{B,1,2}$, R^2_{adj} = 0.921 RMSE = 20.9 s.

	Estimate	SE	tStat	pValue
(Intercept)	2.3081	2.1605	1.0683	0.288
$SCQT_{i-1}$	1.2665	0.037312	33.944	5.50E-56

Table B.12: Regression meta-model for $SCQT_i^{B,1,3}$, R^2_{adj} = 0.122 RMSE = 3.86 s.

	Estimate	SE	tStat	pValue
(Intercept)	0.32419	0.39653	0.81756	0.41559
$SCQT_{i-1}$	0.82007	0.21392	3.8335	0.000223

Table B.13: Regression meta-model for $SCQT_i^{C,1,2}$, R^2_{adj} = 0.981 RMSE = 41.1 s.

	Estimate	SE	tStat	pValue
(Intercept)	7.746	4.339	1.7852	0.077323
$SCQT_{i-1}$	1.1598	0.01631	71.111	3.81E-86

Table B.14: Regression meta-model for $SCQT_i^{C,1,3}$, R^2_{adj} = 0.965 RMSE = 14 s.

	Estimate	SE	tStat	pValue
(Intercept)	0.37526	1.4533	0.25822	0.79678
$SCQT_{i-1}$	1.4064	0.026916	52.25	2.29E-73

Table B.15: Regression meta-model for $SCQT_i^{D,1,2}$, R^2_{adj} = 0.979 RMSE = 24.2 s.

	Estimate	SE	tStat	pValue
(Intercept)	4.5202	2.5635	1.7633	0.080975
$SCQT_{i-1}$	1.322	0.019543	67.647	4.62E-84

Table B.16: Regression meta-model for $SCQT_i^{D,1,3}$, R^2_{adj} = 0.744 RMSE = 9.22 s.

	Estimate	SE	tStat	pValue
(Intercept)	0.55608	0.99251	0.56027	0.57657
$SCQT_{i-1}$	1.2611	0.074305	16.972	6.19E-31

Table B.17: Regression meta-model for $SCQT_i^{A,3,2}$, R^2_{adj} = 0.956 RMSE = 4.75 s.

	Estimate	SE	tStat	pValue
(Intercept)	-0.5956	0.48339	-1.2321	0.22085
$SCQT_{i-1}$	0.56966	0.01233	46.201	2.48E-68

Table B.18: Regression meta-model for $SCQT_i^{A,3,3}$.

	Estimate	SE	tStat	pValue
(Intercept)	0	0	NaN	NaN
$SCQT_{i-1}$	0	0	NaN	NaN

Table B.19: Regression meta-model for $SCQT_i^{B,3,2}$, R^2_{adj} = 0.970 RMSE = 4.53 s.

	Estimate	SE	tStat	pValue
(Intercept)	-0.32839	0.46127	-0.71192	0.4782
$SCQT_{i-1}$	0.65365	0.011559	56.549	1.26E-76

Table B.20: Regression meta-model for $SCQT_i^{B,3,3}$, R^2_{adj} = 0.200 RMSE = 0.02 s.

	Estimate	SE	tStat	pValue
(Intercept)	0.002261	0.00225	1.0049	0.3174
$SCQT_{i-1}$	-0.0114	0.080985	-0.14076	0.88835

Table B.21: Regression meta-model for $SCQT_i^{C,3,2}$, R^2_{adj} = 0.907 RMSE = 53.8 s.

	Estimate	SE	tStat	pValue
(Intercept)	-2.4862	5.9517	-0.41773	0.67706
$SCQT_{i-1}$	0.8489	0.027287	31.109	1.38E-52

Table B.22: Regression meta-model for $SCQT_i^{C,3,3}$, R^2_{adj} = 0.07 RMSE = 0.572 s.

	Estimate	SE	tStat	pValue
(Intercept)	0.062937	0.05799	1.0853	0.28044
$SCQT_{i-1}$	0.28143	0.096759	2.9086	0.004492

Table B.23: Regression meta-model for $SCQT_i^{D,3,2}$, R^2_{adj} = 0.842 RMSE = 115s.

	Estimate	SE	tStat	pValue
(Intercept)	-3.8346	13.014	-0.29466	0.76888
$SCQT_{i-1}$	0.86419	0.037552	23.013	2.66E-41

Table B.24: Regression meta-model for $SCQT_i^{D,3,3}$, R^2_{adj} = 0.02 RMSE = 12.9 s.

	Estimate	SE	tStat	pValue
(Intercept)	2.0782	1.3126	1.5832	0.11659
$SCQT_{i-1}$	0.074875	0.070555	1.0612	0.29119

Table B.25: Regression performance on validation set compared to the fitted residuals.

	$RMSE_{fitted}$	$RMSE_{gen}$
$SCQT^{A,1,2}$	9.39	13.95
$SCQT^{A,2,2}$	106.00	155.53
$SCQT^{A,3,2}$	4.75	0.81
$SCQT^{A,1,3}$	2.90	2.15
$SCQT^{A,2,3}$	8.65	9.62
$SCQT^{A,3,3}$	0.00	0.00
$SCQT^{B,1,2}$	20.90	8.42
$SCQT^{B,2,2}$	82.70	92.32
$SCQT^{B,3,2}$	4.53	0.33
$SCQT^{B,1,3}$	3.86	1.44
$SCQT^{B,2,3}$	9.25	35.12
$SCQT^{B,3,3}$	0.02	0.00
$SCQT^{C,1,2}$	41.10	42.15
$SCQT^{C,2,2}$	171.00	272.94
$SCQT^{C,3,2}$	53.80	2.49
$SCQT^{C,1,3}$	14.00	13.79
$SCQT^{C,2,3}$	31.20	31.73
$SCQT^{C,3,3}$	0.57	0.06
$SCQT^{D,1,2}$	24.20	17.80
$SCQT^{D,2,2}$	135.00	120.04
$SCQT^{D,3,2}$	115.00	77.22
$SCQT^{D,1,3}$	9.22	10.88
$SCQT^{D,2,3}$	36.80	42.12
$SCQT^{D,3,3}$	12.90	37.73

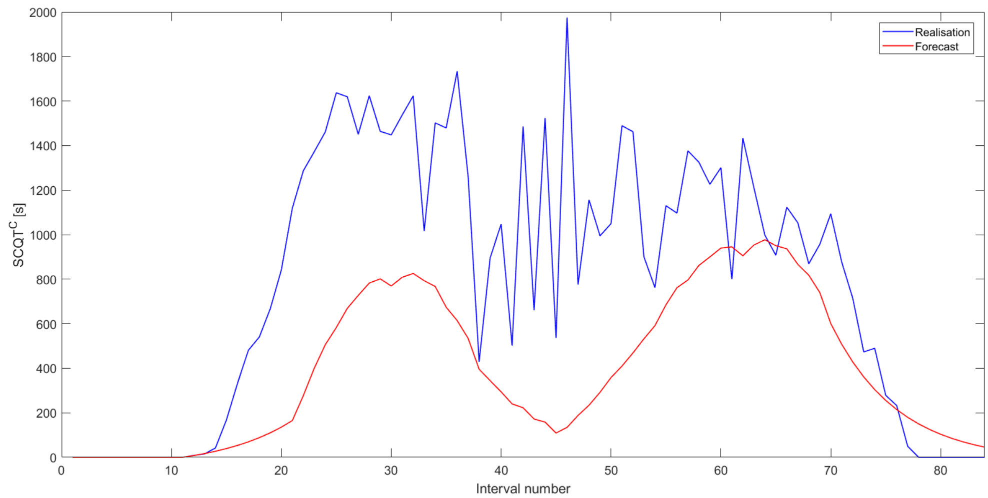


Figure B.3: Security checkpoint C queue time observed and regression model forecast made for a schedule (RMSE = 580 s).(Blue) The realised queue times per time interval. (Red) The forecast of queue times per time interval.

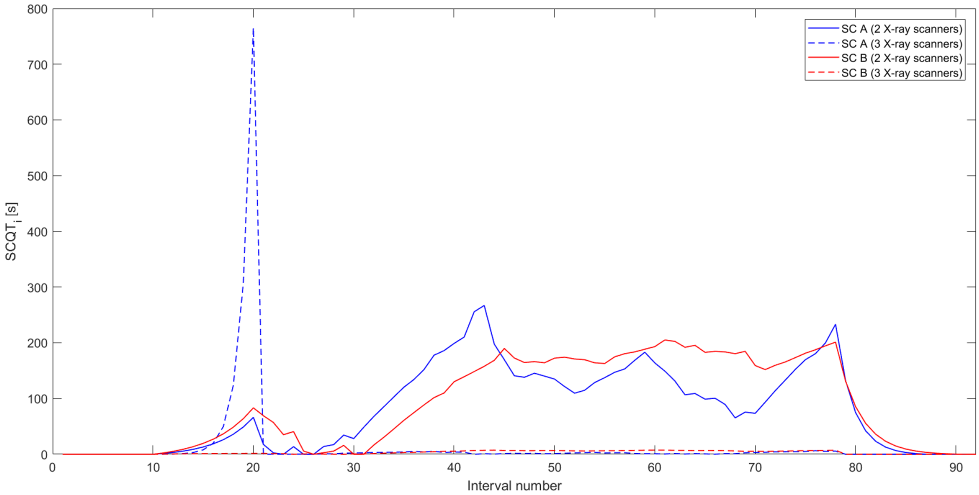


Figure B.4: Forecast queue time Security checkpoint A and B using 2 and 3 X-ray scanners using regression models.

C
Appendix

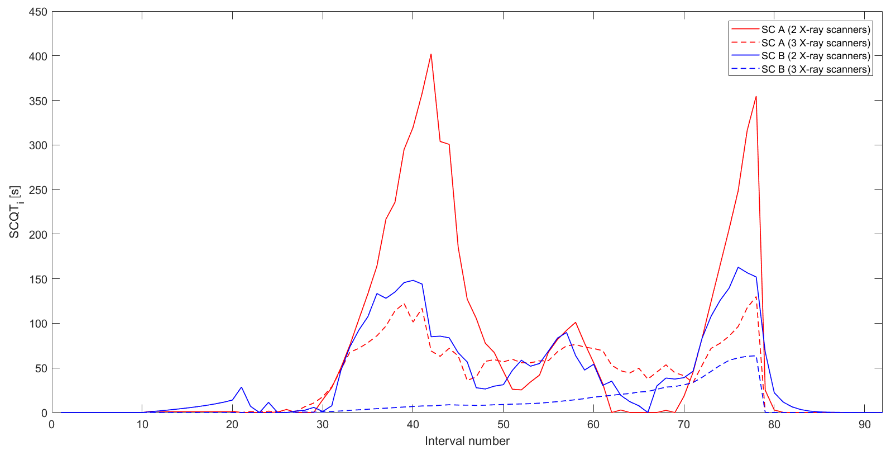


Figure C.1: Forecast queue time Security checkpoint A and B using 2 and 3 X-ray scanners using GRBF models.

Table C.1: Fitting and generalisation errors for potential GRBF models for $SCQT^{D,2,2}$.

	(σ, m) pairs		RMSE [s]	
RBF no	σ	m	Fitting	Generalisation
I	0.1	242	1.35	662.13
II	0.3	150	58.74	255.19
III	0.5	85	108.27	162.95
IV	0.7	52	118.02	144.82
V	0.9	36	126.38	109.79
VI	1.1	27	122.68	123.24
VII	1.3	21	124.31	103.14
VIII	1.5	16	128.2856	109.5064
IX	1.7	13	127.8327	112.1566

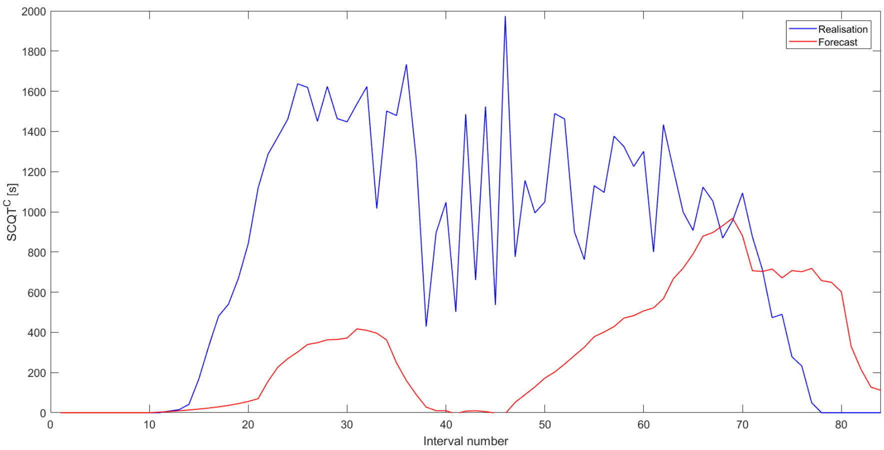


Figure C.2: Security checkpoint C (2 X-ray scanners) queue time observed and GRBF model forecast made for a schedule (RMSE = 779 s).(Blue) The realised queue times per time interval. (Red) The forecast of queue times per time interval.

D
Appendix

Table D.1: Differential evolution algorithm settings for the validation case study.

Parameter	Value	Unit
Population size	100	Ind.
Stopping condition tolerance	1.00E-05	[-]
Total generations	5000	It.
Penalty minimum	60	[-]
Penalty maximum	110	[-]