TU Delft Delft University of Technology

nlr Dedicated to innovation in aerospace

Master Thesis Project | April 2018

Autonomous Transport Robots in Baggage Handling Systems

A study on the use of autonomous individual transport robots in baggage handling systems at medium-sized regional airports operating in a point-to-point network

A.A.E. Silven | 4251563

Autonomous Transport Robots in Baggage Handling Systems

Using a small-scale agent-based model to evaluate a new baggage handling concept

A.A.E. (Alisa) Silven
Student number: 4251563

Master thesis for MSc. in Transport, Infrastructure and Logistics
Delft University of Technology

Report number: 2018.TIL.8218
Date of submission: April 6th, 2018

Graduation Committee:

Prof. dr. R.R. Negenborn	Transport Engineering and Logistics, Faculty of 3mE
Dr. J.H.R. van Duin	Engineering Systems and Services, Faculty of TPM
Dr. ir. Y. Pang	Transport Engineering and Logistics, Faculty of 3mE
ir. R.J.D. Verbeek	Dutch Aerospace Centre, ATM and Airports

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

Cover photo: Airport sign picture, via https://unsplash.com

This page is intentionally left blank

Summary

The growing international air travel demand implies capacity challenges for airports worldwide. Conventional baggage handling systems are a critical element of airport operations determining the handling capacity of the whole airport. Baggage handling systems are currently unable to properly adapt to demand fluctuations in the aviation industry. By investigating the state of the art of both baggage handling and transport robot systems, the Baggage Robot Concept is introduced, which combines the two systems. The proposed Baggage Robot Concept uses autonomous and individual transport robots to make the floor plan and desired capacity of airport baggage handling systems more dynamic.

The floor layout of the baggage handling area can be adjusted easily, because the Baggage Robot Concept uses electromagnetic induction in the floor of the baggage handling area to charge the batteries of the robots. This also means that the charging infrastructure does not constitute any obstacle for transport robots in motion. The transport robots can thus transport bags over the shortest paths possible between the entrances and exits of the baggage handling system. When a number of robots have to perform their transport task simultaneously, there is a risk of collision between robots. Two collision avoidance measures are defined to prevent actual collisions: (i.) stopping and resuming policy: in case of an imminent side collision, (ii.) turn, (wait) and continue policy: in case of an imminent frontal collision.

By invoking the collision avoidance measures when collisions are imminent, deadlocks are avoided too. To realize abovementioned elements in the Baggage Robot Concept, a hybrid control architecture is identified to be most suitable. The hybrid control is an aggregation of a decentralized control architecture necessary to provide the individual robots with decision authority and a centralized control architecture necessary to improve the performance of the system as a whole. This hybrid control architecture is suitable for the Baggage Robot Concept as it increases the robustness, scalability, flexibility and performance of the system.

To evaluate the performance of the Baggage Robot Concept, key performance indicators were formulated. When the Baggage Robot Concept performs as well or even better than the conventional baggage handling system, the Baggage Robot Concept is found to be a feasible addition to or replacement of a conventional baggage handling system. Under the assumption that transport robots provide most value in the sorting process, an agent-based simulation model is developed to test the performance of the sorting process of this Baggage Robot Concept on a medium-sized airport operating in a point-to-point network.

By implementing the Baggage Robot Concept, a step is made in dynamically altering the floor plan and desired capacity of airport baggage handling systems. The performance of this Baggage Robot Concept has been reflected upon using results from the simulation model. Exact design parameters or values for these parameters cannot yet be given due to the limitations of the developed simulation model and the lack of reliable data available for a far-future concept. Yet, the number of robots and the layout configuration were found to be the most important design parameters. The number of robots should be such that the average process time of bags does not significantly decrease when introducing an additional robot to the system. The layout should be such that the route the robots travel is as short as possible, while at the same time ensuring the robots have enough manoeuvre space to reduce the number of possible conflicts. Depending on the number of bags to be handled and the arrival pattern of these bags, a different minimum number of robots is necessary. When the largest emphasis is put on the performance of the Baggage Robot Concept in terms of the percentage of mishandled bags, more robots are needed than strictly necessary to handle all incoming bags in time. Depending on the interest of the owner of the Baggage Robot Concept, the importance of decisions on acquisition costs or performance can shift.

Future research should investigate the success of the Baggage Robot Concept as a supplement to or a replacement of conventional baggage handling systems. A suggestion for future research is to explore the technical feasibility of the proposed robot types. One important starting point can be to research the sensing capabilities of the transport robots to gain insight into the technical possibilities of these robots. To do so, inspiration can be taken and lessons can be learned from innovations in the field of autonomous cars. Although the autonomous car technology is still being developed, it can help to improve the individual transport robot systems. In this future research, timescale and cost estimations for research and development can help in estimating on which timescale the proposed baggage robot concept could be built into a prototype system to further test it.

This thesis has been a first exploration in integrating autonomous robot systems in baggage handling systems, contributing to future proof and cost efficient airport operations.

Table of Contents

1. Introduction

People want to fly. Not only do people want to fly, the air transport industry provides vital economic benefits and can be considered the engine of global socio-economic growth (The World Bank, 2017). By creating direct and indirect employment, supporting tourism and local businesses, and stimulating foreign investment and international trade, the industry is of great importance for economic development (ATAG, 2005). When it comes to business and leisure air travel, people travel with their belongings which can be checked-in to a flight as well. These checked-in belongings – referred to as checked-in baggage – play the central role in this research. This chapter starts by exploring the growing air transport industry which leads to the research problem as this growth has implications on the capacity of airports.

1.1 Growing Air Travel Demand

Ever since the first jet airliner took off in 1949, the use of commercial aviation has grown. No other major transport mode yet was able to match this growth. The growing demand for air travel influences the global economy by making it possible for millions of people and billions of dollars' worth of goods to rapidly move around the world (ATAG, 2005). This research focuses on passenger air travel from civil airports.

Data from The World Bank shows the total scheduled traffic carried by all air carriers worldwide, both international and domestic. The source for this data is the International Civil Aviation Organization, abbreviated to ICAO. The data shows that since 1970, the number of passengers carried worldwide has increased from around 310.4 million in 1970 to almost 3.7 billion in 2016. In this period, the growth trend has continued with an average of 5.76% per year, with a maximum of 16.8% in 2010 and a minimum growth of 0.82% in 1998. The trend upwards is also visible in Figure 1, in which a few irregularities are found, explained by the six years that show a decrease in the number of passengers carried, with a low of -1.69% in 2002 (The World Bank, 2017).

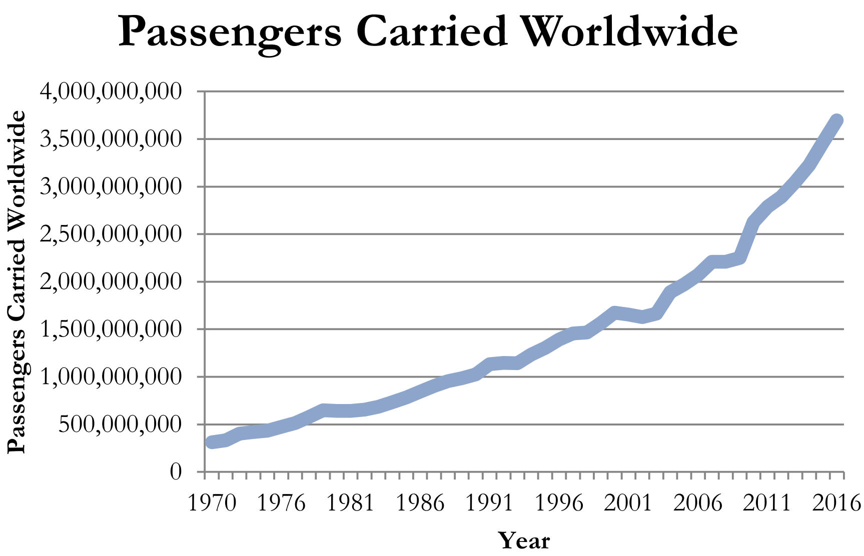

Figure 1 - Passengers Carried Worldwide in Civil Aviation, data from (The World Bank, 2017)

Not only the past shows an upward trend in the popularity of air transport as a means of travelling, significant growth in the future is also expected. The International Air Transport Association (IATA) keeps statistics on airline traffic data. IATA expects the yearly number of air travellers to roughly double in the period 2016-2035, from 3.8 to 7.2 billion. In her two-decade forecast, IATA presents three possible scenarios. The central scenario – the blue line in Figure 2

– foresees the doubling of passengers to 7.2 billion in 2035, under the assumption of an annual compound average growth rate (CAGR) of 3.7%. However, a decrease of the CAGR to 2.5% is expected if the current trend towards trade protectionism gets stronger and continues in the long term. This boils down to 5.8 billion air travellers in 2035 – the green line in Figure 2. The final scenario is the opposite of the protectionist scenario. In this scenario – the pink line in Figure 2 - trade liberalisation is assumed. This scenario implies a growth in demand up to three times the 2015 level (IATA, 2016). This 20-year forecast shows that growth is expected in every scenario and the aviation industry needs to be ready to deal with it.

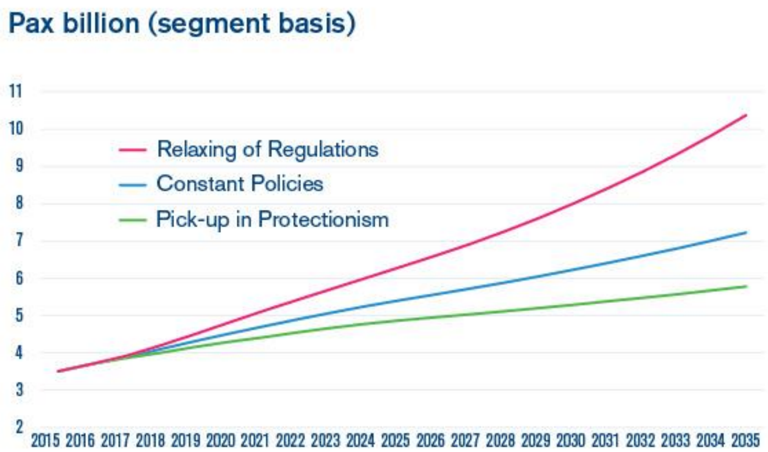

Figure 2 - Forecast Scenarios (IATA, 2016)

The demand for air travel shows a growing trend. However, these yearly total numbers do not show variations in demand over time within a year. By looking at variations in demand in the course of a year, the challenge of demand management is illustrated. Like all parties in the tourism industry, airports have to deal with major fluctuations in demand. Airports popular among tourists are more dependent on leisure demand and have to deal with larger differences between peak and off-peak periods. The growing popularity of low-cost carriers and the gradual shift in holiday preferences to more frequent short breaks flatten the peaky curve of leisure demand somewhat.

Figure 3 shows the numbers of passengers (in thousands) handled in the top 30 airports in Europe per month in 2016. The peak at most European airports is in the July-September period, when all countries have their summer breaks. The changing demand patterns over the months are influenced by tourism and the combination of airline and passenger profiles. These fluctuations indicate how airport infrastructures are utilised during the year and that handling peak flows must be incorporated in the design of these infrastructures. This implies that when the airport is able to handle peak flows in peak periods, it also has to deal with the under-utilisation of the infrastructure during off-peaks, leading to an inefficient operational cost structure (Papatheodorou, Graham, & Forsyth, 2016).

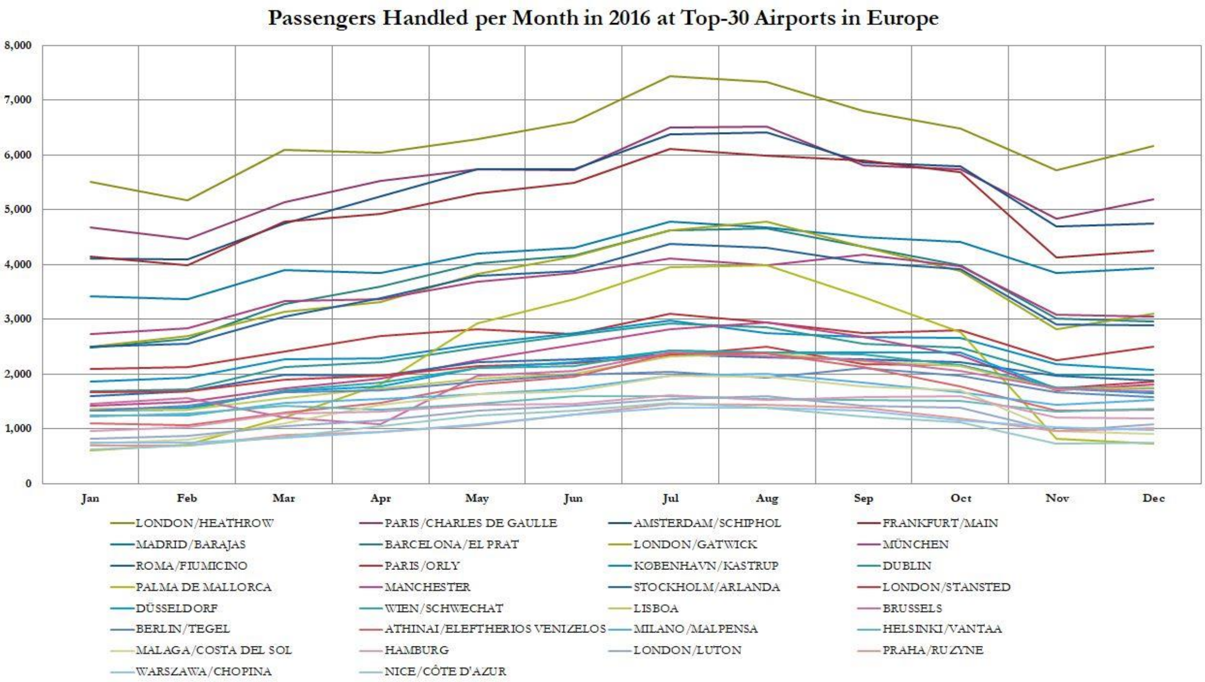

Figure 3 - Overview of thousand passengers handled in top 30 airports in Europe on a monthly basis. Data from Eurostat (Eurostat, n.d.)

Changing an entire baggage handling system to cope with increasing demand is not an easy job. A striking example can be found at Denver International Airport. In the 1990's the airport realized that baggage handling is a critical element in an airport as big as Denver's (Montealegre & Keil, 2000). They decided to change the conventional baggage handling system into an automated system, which was supposed to be the world's largest automated airport baggage handling system, strengthening Denver's position as an air transport hub. By automating the baggage handling, the turnaround time was to be reduced to only 30 minutes. However, the Denver project became a well-known story on how technology projects can go wrong. Denver airport suffered from a combination of many failures such as the underestimation of the complexity of the project, a lack of planning, poor design, people working on their own part without communicating with other relevant parties and a lack of risk management. The lack of knowledge within the construction party combined with repeatedly ignored expert advice was the main cause of the project resulting in a failure (Calleam Consulting Ltd, 2008).

1.2 Research Problem: Implications of Growing Air Travel Demand

International air travel demand has been increasing for many years now and forecasts show that the growth in air travel will also be substantial in the future (Graham & Metz, 2017). This implies challenges for airports worldwide because main airports have difficulties matching their current capacity with the ever-increasing air travel demand (Mota & Boosten, 2011; Sun & Schonfeld, 2015). This is true for growing demand on a yearly basis and also for fluctuating demand on a monthly or seasonal basis. At airports worldwide elements such as runways, terminals, air traffic control and more need to be expanded to cope with the growing numbers of passengers (IATA, 2016). One of the critical elements determining the capacity of an airport is the baggage handling system (Cavada, Cortés, & Rey, 2017). A traditional baggage handling system, or BHS, consists of a complex system of conveyor belts with many connected elements. The rigid nature of these conveyor belts makes the system very laborious and lengthy to implement, and also incapable of adapting to changing circumstances. It is not an easy job to relocate the capacity of the BHS at the airport, neither is it possible to easily scale the systems up or down. This, in combination with the long-term uncertainty of demand, makes it impossible for the conventional baggage handling system to properly adapt to the seasonal demand peaks of the aviation industry. Currently, baggage handling systems are 'oversized by design': they have spare capacity to be able to handle growth in passenger numbers and to allow for failures in parts of the system. This prevents the need for alterations to the implementation as much as possible, but comes with an undesirable cost as a result of having overcapacity for many years. This leads to a need for research that focuses on the development of a completely new baggage handling concept at airports, to cope with these challenges and to be less dependent on the rigid conveyor belt systems. This new baggage handling concept must provide 'dynamic capacity' and eliminate as much as possible the need to invest in overcapacity that will be not be used for a number of years.

1.2.1 Improving Baggage Handling Systems through Dynamic Capacity Solutions

The idea for a completely new way of transporting baggage over an airport has been put forward by a European research project, called the 2050+ Airport, which is sponsored by the European Commission (European Commission, n.d.). The research group envisions a future airport on which small robots transport individual baggage items across the airport on a point-to-point basis. By doing so, the baggage handling process could become more flexible and simple, but this comes at a cost. Taking into consideration the forecasts showing that the growth in air travel will also be substantial in the future (Graham & Metz, 2017), the need for research on this concept is born.

An example of this new way of transporting baggage that seems interesting to research is the so-called BagBot concept. BagBot is a concept based on the use of individual transport robots that allows a baggage handling system to be implemented in a flexible and scalable way. These robots provide point-to-point transport of baggage items and have the benefit that they offer flexibility, scalability and (re)prioritization possibilities, due to the autonomous nature of robot units. In this concept, the robots replace the conventional conveyor belts in the baggage handling system. Applying these robots in a BHS environment should, in theory, enable the BHS to accommodate the growth or seasonal needs of an airport.

1.3 Research Goal and Methods

The situation of growing air travel demand and its implications has led to the urge of finding a new way to dynamically alter the capacity of baggage handling systems. This research has the goal of investigating the use of autonomous and individual transport robots in baggage handling systems. The next sections are concerned with the research questions, the scope, the research methodology and methods and ends with the structure of this thesis.

1.3.1 Research Questions

Following from the research problem, a need for a research that focuses on the feasibility of autonomous and individual transport robots concept applied to BHS becomes apparent. However, this concept has never been applied to a baggage handling system. This is why the main research question is formulated as follows:

In what way is it feasible to dynamically alter the floor plan and desired capacity of airport baggage handling systems by making use of autonomous individual transport robots?

To answer this main question, the following sub-questions are formulated:

1. What is the current state of baggage handling systems and transport robots?
2. What Key Performance Indicators (KPI's) are relevant in assessing the capacity and continuity of a baggage handling system that makes use of autonomous individual transport robots?
3. How can the autonomous individual transport robot concept be used in baggage handling systems?

4. How can the performance of a baggage handling system that makes use of autonomous individual transport robots be predicted and evaluated?
5. What does the performance of a baggage handling system with autonomous individual transport robots look like?

To answer the research questions, the scope and research methods need to be made clear.

1.3.2 Scope

The main research question can be cut into four main parts: In what way is it **feasible** *(1)* to **dynamically alter the floor plan and desired capacity** *(2)* of **airport baggage handling systems** *(3)* by making use of **autonomous individual transport robots** *(4)*?
These parts delineate the scope of this research.

(1) Feasible: is it technically feasible to implement autonomous individual transport robots in an airport baggage handling environment? In other words, is the concept possible and practical in this environment? Other feasibility perspectives such as environmental, economic or legal feasibility are left out of the scope of this research.
(2) Dynamically alter the floor plan and desired capacity: this part focuses on the ability of the concept to adapt to changing demand in 'real-time'. If the system is flexible it will be able to relocate its capacity in or between airports or scale up or down to adapt to seasonal or event-related demand peaks. The dimensions of the space in which the baggage handling has to take place are considered constant being the size of a medium-sized regional airport, operating in a point-to-point network, comparable to Eindhoven Airport in the Netherlands. The fixed space size however does not mean that the layout of the floor plan is necessarily fixed. Different floor plans and the effect of different floor plans on the performance indicators are considered in this research.
(3) Airport baggage handling systems: currently baggage handling systems on airports are complicated systems of conveyor belts with many connected elements, which are complex and laborious to implement. A baggage handling system transports checked bags from bag drop-off facilities to makeup stations were bags are loaded onto baggage carts that transport the bags to aircraft. At some airports, the baggage handling system also transports unloaded bags form arriving flights to the baggage reclaim area. This reversed process is not included in this research. This research mainly focuses on the sorting task of the baggage handling system and therefore puts less emphasis on security screenings and makeup. As the scope is a medium-sized regional airport that operates in a point-to-point network, transfer baggage is excluded. Odd-sized baggage is also not included in this research, as odd-sized baggage is not handled in the regular sorting system but dealt with in a manual process. Only bags that weight less than 32 kilos and have dimensions of maximum 100 cm in length, 65 in height and 75 cm in width are considered. For the construction of the simulation model, only the sorting element of baggage handling systems is modelled. This element is of most interest in this research, as it is assumed that transport robots provide most value in the sorting process.
(4) Autonomous individual transport robots: For the use of individual transport robots, autonomous transport robots are the focus. These robots can carry one baggage item at the time and only bags or suitcases are considered as baggage items in this research. The degree to which it is possible to include autonomy in a baggage handling system that makes use of transport robots is discussed in this research.

1.3.3 Research Methodology and its Connection to the Research Questions

A well-supported design science research methodology as formulated by (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007), helps to structure this research. In their paper the authors have outlined their new methodology for information systems, as no methodology existed for problems including both IT and organizations. Since their methodology can be applied to research *new* technologies within *existing* organizations, it suits this research with its focus on the feasibility of a new technology – autonomous transport robots – in an existing environment, the baggage handling area of medium-sized regional airports. This section shows how this methodology supports the research questions and methods as discussed in section 1.3 and section 1.3.4 respectively.

The design science research methodology consists of six steps:

1. Problem identification and motivation
2. Definition of the objectives for a solution
3. Design and development
4. Demonstration
5. Evaluation
6. Communication

These steps are briefly described here.

1. Problem identification and motivation is about identifying a clear problem and motivating why this problem is relevant to solve. From section 1.1 it became clear that the demand for air travel has been growing in the past and is expected to keep growing in the future. This growth, driven by factors such as rising GDPs, reduced air travel costs, globalisation and deregulation (ATAG, 2005), has great benefits on different aspects but it also has its implications. Section 1.2 describes the main implication that this research is about, which is conventional baggage handling systems being 'oversized by design' to have spare capacity to be able to handle growth in passenger numbers in the future and to allow for failures in parts of the system. The undesirable cost as a result of having this overcapacity for many years is the motivation to research alternative systems that enable dynamic capacity adjustments. The 2050+ Airport research project, supported by the European Commission, underlines this motivation by proposing to use small transport robots. The first and third research questions aim to provide a deep understanding of both conventional baggage handling systems and small transport robots to investigate how these transport robots can help in making the capacity of baggage handling systems more dynamic.
2. The second step – definition of the objectives for a solution – follows from the first step. As the problem is identified and described, the objectives of the answer to the problem need to be defined. In this research, the problem – undesirable overcapacity in conventional baggage handling systems for years – will be addressed by the design of a new system – a baggage handling system where autonomous and individual transport robots perform the sorting of bags. To investigate if this new system is a feasible solution to the research problem, objectives, i.e. requirements for this solution need to be identified. The second research question is addressed this step. This question is about the identification of key performance indicators useful for assessing the performance of the solution, as well as the identification of requirements, both functional and non-functional. The solution system should be able to meet these requirements and answer the key performance indicators, while complying with the applicable constraints.
3. The third step – design and development – is about 'creating the artefact'. The artefact in this research is the simulation model of autonomous transport robots performing the sorting task in a baggage handling system. This third step includes 'determining the artefact's desired functionality and its architecture and then creating the actual artefact' (Peffers et al., 2007). For this research, the design and development phase of the simulation model is supported by Dam, Nikolic, & Lukszo (2010). The outcome of this step answers the fourth research question.
4. The fourth step – demonstration – concerns the use of the simulation model designed and developed in the third step. According to Peffers et al. (2007) demonstration of the solution can serve multiple purposes, from proving that the solution works to a more formal

evaluation of the developed artefact. In this research, the simulation model will be used to demonstrate the use of autonomous and individual transport robots in the sorting process of baggage handling at airports. Experiments with this model will show if these robots are useful in executing the sorting task and can be a start of proving that deploying these robots can be a solution to the problem of having overcapacity for many years.

5. The fifth step – evaluation – also uses experiments in the simulation model. However, by using the key performance indicators formulated in the second research question, an evaluation of the solution to the problem can be performed. From running experiments and generating performance data from the simulation model, observations and measures can show to what extent the autonomous and individual transport robots provide a solution to the research problem. An evaluation of the use of autonomous and individual transport robots as being a feasible solution to the problem is the outcome of this evaluation step. It is also the answer to research question five.
6. The sixth and final step – communication – is about communicating the research problem, the importance of solving the research problem and the developed simulation model supporting the solution to the problem. As this research is executed in partial fulfilment of the requirements for the degree of Master of Science, this communication step mainly involves communicating the research findings and the proposed solution to parties involved by means of this master thesis report.

1.3.4 Research Methods

The research flow diagram depicted in Figure 4 shows the research methods used for answering the sub-research questions leading to the answer of the main research question.

To answer the first research question, the conventional baggage handling system needs to be investigated thoroughly, as well as the use of transport robots in other industries. The first part – investigating the current state of baggage handling systems – will be done by means of literature research in combination with conducting an expert interview with a Dutch baggage handler and a site visit at a medium-sized regional airport in the Netherlands, Eindhoven Airport. A combination of these three research methods leads to a deep understanding of the conventional baggage handling system at a medium-sized regional airport.

The second part – investigating the current state of transport robot systems – will be done in a similar way. The most important research method here is literature research as transport robot systems are implemented in other industries like warehousing and distribution. A site visit to a parcel and e-commerce corporation in the Netherlands, an interview with experts from this corporation and an interview with an expert from an independent research organisation provide the input to finalize the answer to this first research question.

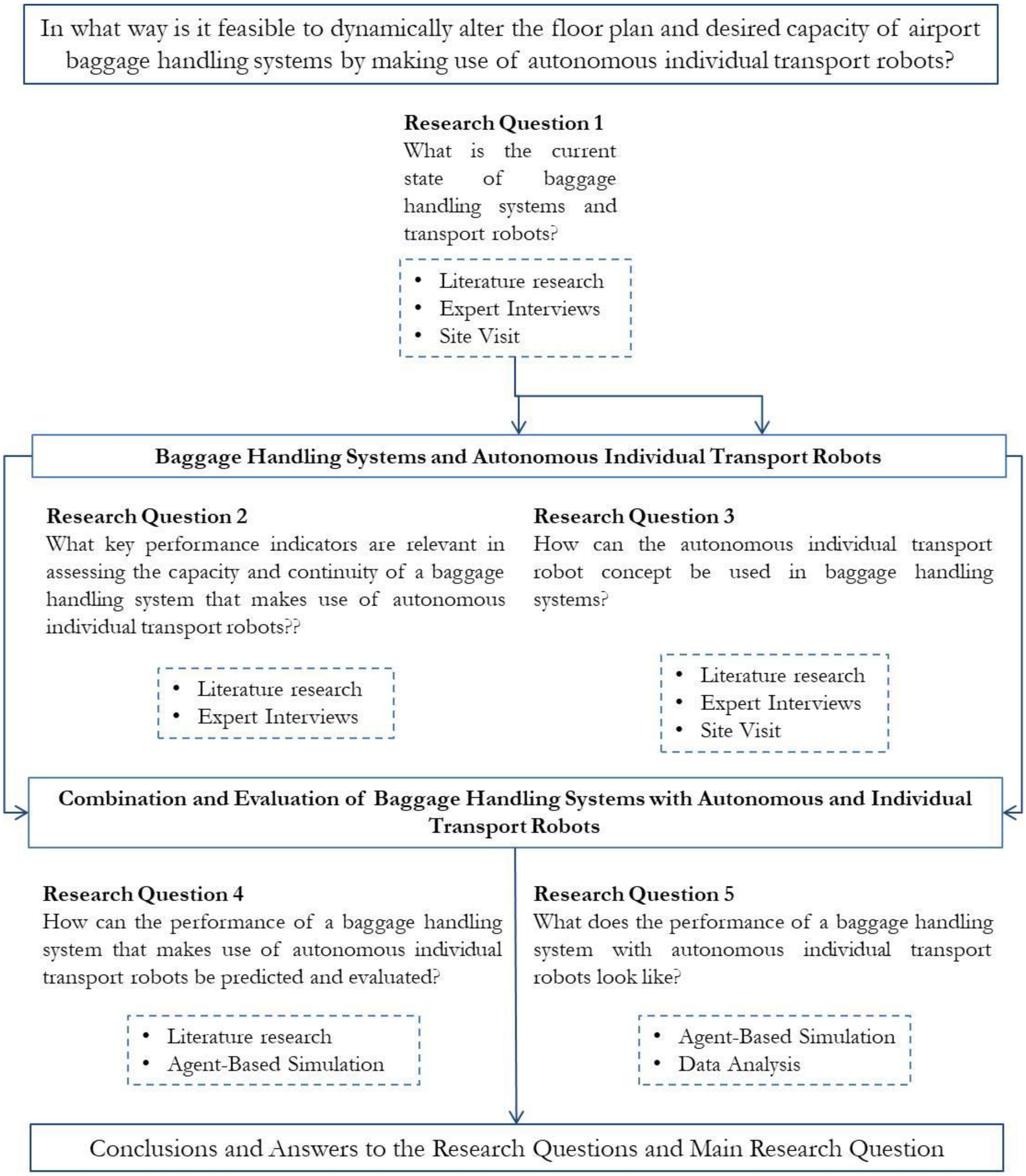

Figure 4 - Research Flow Diagram

The second research question is focused on baggage handling systems. Performance indicators need to be identified to properly assess the performance, capacity and continuity of a baggage handling system. The identification of these key performance indicators will follow from a combination of literature research and expert interviews with a baggage handling organization. This expert interview focuses on performance indicators of conventional baggage handling systems, whereas the literature research will focus on a combination of performance indicators for conventional baggage handling systems and transport robot system.

The third research question investigates where in a baggage handling system transport robots can be of most value. This will be done by means of site visits and expert interviews at the same organizations and with the same people as visited and interviewed for the first research question.

Combined with literature research on Automated Guided Vehicles (AGV) – a typical type of transport robots – this will lead to a thorough understanding of AGV-systems that are already operational, which will help by placing the advancement of the transport robots concept into a baggage handling perspective.

Answering the fourth research question starts with a literature research to make a well-founded choice for an appropriate simulation method. This method will then be used to simulate the autonomous transport robot concept. The development of the simulation model will include an elaboration on several relevant algorithms used to correctly represent autonomous transport robots in a baggage handling environment.

The fifth and final research question uses the performance indicators as formulated in the second research question and the simulation model following from the fourth research question to evaluate the autonomous transport robot concept in a baggage handling context.

1.3.5 Thesis Structure

The structure of this thesis follows from the research methodology elaborated on in section 1.3.3. Chapter 2 explores both conventional baggage handling systems and transport robot systems, providing context. This chapter attempts to identify where in the baggage handling process transport robots can be of most use in resolving the problem of inflexible capacity, corresponding to the first step in the design science research methodology – problem identification and motivation.

Based on this chapter, a newly designed sorting system for baggage handling is proposed in Chapter 3, including the requirements this design should comply with. This chapter corresponds with the second step of the methodology – definition of the objectives for a solution – and the design part of the third step – design a development.

Chapter 4 builds on the third chapter, as it concerns the development of a simulation model which represents the design of a baggage handling system making use of transport robots. This chapter corresponds with the development part of the third methodology step – design and development.

After the simulation model is developed and verified, Chapter 5 uses the model to demonstrate the performance of a baggage handling system that makes use of transport robots by means of running experiments with the model. Based on the requirements and performance indicators defined in Chapter 3, the performance of the newly designed baggage handling system is evaluated. Chapter 5 corresponds with the fourth step in the methodology – demonstration – and the fifth step – evaluation.

Chapter 6 marks the final chapter of this thesis. In this chapter, the research questions are answered and a critical reflection on the research is included. Based on the results obtained, recommendations on future work are put forward. As mentioned earlier, the sixth and final step in the methodology – communication – involves communicating the research findings by means of this thesis report.

2. Baggage Handling and Transport Robot Systems in Practice

This chapter focuses on the practical situation of conventional baggage handling at airports and the use of transport robot systems – AGVs in particular – in other industries such as warehousing and container terminals. It represents the first research question on the current state of baggage handling systems and transport robot systems.

The first part of this chapter explains conventional baggage handling, to get to a thorough understanding of all the processes present in a conventional baggage handling system typical for a large hub airport. As this research focuses on the baggage handling system at a medium-sized regional airport, a case study on the baggage handling system at a medium-sized regional airport in the Netherlands is performed in section 2.1.1. Section 2.1.2 finalizes the baggage handling system part of this chapter by elaborating on current developments in these systems.

The second part of this chapter focuses on transport robot systems and AGVs in particular. Section 2.2 elaborates on what AGVs are. The most important difference between AGVs and autonomous transport robots is the type of control. Section 2.2.1 describes the control of AGVs, to help understand the key differences between autonomous robots and AGVs later on. The second part of the chapter concludes with an exploration on the use of AGVs in warehouses and container terminals in section 2.2.2 and section 2.2.3 respectively.

With a thorough understanding of both baggage handling systems and transport robot systems this chapter concludes with a synthesis, combining the flexibility advantage of transport robot systems with the inflexibility disadvantage of baggage handling systems.

2.1 Conventional Baggage Handling System in Practice

The process of baggage handling starts as soon as a passenger checks in his/her baggage item(s) at the airport. This can be done simultaneously with checking in for a flight, but this is not necessary. Nowadays, more and more people check in for their flight before they arrive at the airport, using either a computer or a mobile phone. Another check-in option is to use a self-service check-in kiosk, present at most airports. When a passenger has obtained a boarding pass, and wants to drop off baggage items, the passenger can proceed to a bag drop facility, either at a check-in desk (with help of an airline employee) or at a self-service bag drop facility. After scanning the boarding pass, a baggage label is printed that must be attached to the baggage item. This baggage label – also referred to as baggage tag or luggage ticket – contains information on the airline name, flight number and destination airport code and includes a bar code that can be read by hand held scanners and scanners that are part of baggage conveyor systems. At the drop off, characteristics such as weight and dimensions of the baggage item are registered, but information on these characteristics is usually discarded right after the process of checking in. This information is therefore not accessible and used in the remainder of the baggage handling process. After successfully attaching the label to the baggage items, the baggage items proceed to the baggage handling area and the passenger can proceed to the security check.

Once the baggage items reach the baggage handling area of the departure airport, five processes take place in order to get the right baggage items on the right aircraft:

(1) performing security checks on the content of the baggage items
(2) sorting of baggage items
(3) loading sorted bags on baggage carts at the makeup stations
(4) transporting the baggage items to the right aircraft
(5) loading the baggage items into the aircraft.

In case there is more than one baggage handler present at the departure airport, the baggage label shows which ground handler the baggage item is assigned to in the handling process (Swissport, personal communication, February 22, 2017).

Once the labelled baggage items enter the baggage handling area, the first process is to get them cleared by means of security screenings. At most international airports, five screening levels are used. With each screening level the inspection becomes more detailed.

- Level 1 screening consists of an x-ray scanner that automatically examines if the baggage item can be cleared or needs further inspection. If the item is cleared after level 1 screening it proceeds to the sorting area, if not the item is not cleared and proceeds to screening level 2.
- Level 2 screening requires a trained human operator examining the x-ray picture taken in level 1. If necessary, the operator can make a new x-ray picture to re-examine the content of the baggage item. If the operator clears the item, it continues to the sorting area, if not it proceeds to screening level 3.
- At level 3 screening a CTX picture is taken, more detailed than an x-ray picture as the picture of the bag is sliced into multiple pieces. The CTX-machine is capable of accepting or rejecting a baggage item. In case the machine accepts the item, it continues to the sorting area, but when the machine rejects the item a trained operator must analyse the CTX picture.
- This human inspection of the CTX picture is the screening level 4. If the baggage item does not pass the human inspection, the item will be removed from the system and the owner of the baggage item is called.
- Level 5 screening is the manual investigation by a trained human in the presence of the bag's owner (Grigora & Hoede, 2007).

Figure 5 visualises the security screening process and its place in the baggage system as a whole. Orange stars indicate the mandatory tracking points for baggage items, resulting from IATA Resolution 753. The resolution becomes effective on June 1st, 2018 and is intended to encourage airlines to further reduce baggage mishandling (IATA, n.d.).

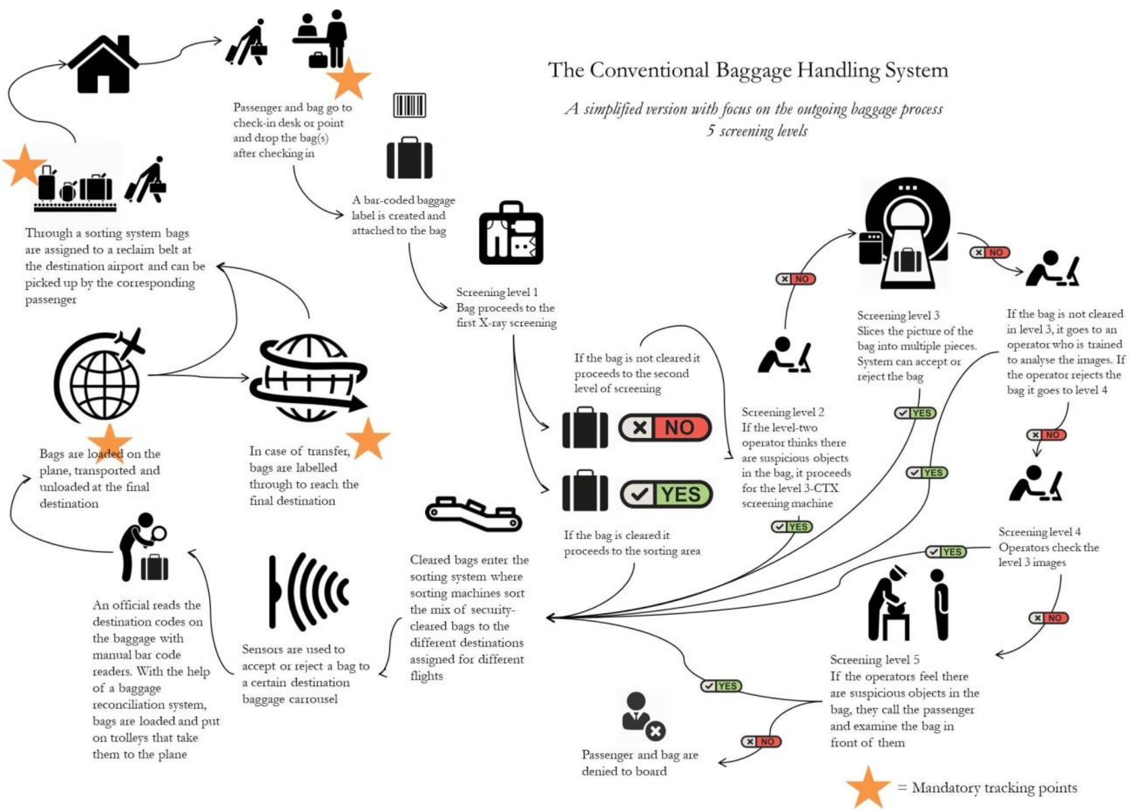

Figure 5 - The Conventional Baggage Handling System, focus on security screening (Image Author, 2017)

As soon as the baggage item has cleared the security screening, conveyor belts lead it through a tag reader where the baggage label is read. The automated sortation system reads the information on the label and uses it to route the item directly to the correct so-called ‘makeup area’. This routing is the sorting process. However, some passengers check their baggage items more than three hours before departure. These baggage items are stored in an Early Bag Storage – EBS – which is available on most (not all) airports. In traditional baggage handling systems, workers have to manually remove the early arrived items from the system and transport them to the EBS. This method of relocating baggage items to the EBS has its disadvantages. Manually moving baggage items around increases the amount of work related injuries, resulting in a secondary effect of higher labour cost. However, it is not just labour costs that increase in a manual system. Since the early items are moved and stored manually, managing and tracking these items – including ensuring they are properly stored and returned to the sortation system in time – becomes the responsibility of the worker. Since workers in baggage handling areas work in shifts, accurate handling of early arrived items is complicated and the risk of human error increases. Failures in this system of manually keeping track of and transporting baggage items lead to greater chances for mishandled baggage, leading to additional cost for the airline (Dadyala, 2013).

If the handling of early arrived bags is done accurately, all early bags are back in the sorting area in time. In the sorting area, the conveyor belts constantly receive and release bags to sort destinations. Fully automated sorting machines sort the mix of security-cleared baggage items to the different destinations assigned to different flights. A 270 degrees infrared sensor on the baggage carousel in the sorting area reads the baggage labels and sends the destination information to a processor. This processor uses an algorithm that is able to make a decision in microseconds to accept or reject a baggage item for the different destination baggage carrousels.

Once the baggage items are sorted by the baggage handling system, a worker uses a manual bar code reader to read the destination codes on the baggage items. When the baggage items are scanned, a reconciliation system helps the workers to load the bags on wheeled platforms. Then the bags can be put on trolleys that take the baggage items to the aircraft or to a storage area. Not all airlines and airports use trolleys. Unit load devices (ULDs) are also commonly used for larger aircraft. These ULDs can be mechanically loaded and unloaded from the aircraft (Odoni, 1980). After the baggage items are inside the aircraft and all corresponding passengers have boarded, the aircraft is ready to take off to its destination airport.

At the destination airport, a reversed process takes place, but much simpler. When the aircraft arrives at the destination airport, all baggage items are removed from the aircraft and transported to the baggage handling system of the airport. In the baggage handling area, the baggage items are loaded in a sorting area, where transfer baggage is separated from baggage items that have reached their final destination (KLM Ground Services, personal communication, March 13, 2017). The baggage items that have reached their final destination are transported to the correct reclaim belt, which can be accessed by arriving passengers in the reclaim area. These incoming baggage items are directly loaded onto the reclaim belt and are therefore no part of the conveyor belt system of baggage handling. Transfer baggage is divided into two types, regular transfer baggage and transfer baggage with short transfer times. The regular transfer baggage re-enters the baggage handling system via the transfer belt. Upon entering the baggage handling system again, the baggage items need to undergo security screening again. However, baggage items that have been cleared at an airport that is declared safe will not be sent to the next level of security. Baggage items that are entering the baggage handling system from other airports will have to pass the regular five level security screening process. For transfers with a short transfer time, baggage items can be transported tail-to-tail, which means the items are unloaded from the arriving aircraft and directly transported and loaded into the connecting aircraft.

For odd-sized baggage items the baggage handling process is different. The term odd-sized refers to baggage items that are larger or heavier than normal, like wheelchairs, bicycles, golf bags or large music instruments (Schiphol Airport, n.d.). Generally, items are considered odd-sized if they are between 32 and 80 kilos or have dimensions exceeding 100 cm in length, 65 cm in height and 75 cm in width. These items can't be handled by conventional baggage handling systems and are therefore handled manually. The steps of the baggage handling process are similar but the consecutive handling steps take more time. Where regular baggage items are transported to the baggage handling area by conveyor belts, odd-sized baggage is transported to this area manually. Since not all odd-sized items can be scanned by the automated x-ray or CTX machines because of weight or size restrictions, these items are separated and immediately go to the fifth security screening level. After the odd-sized item is cleared it is loaded into a unit load device (ULD) to be transported to the aircraft.

2.1.1 Baggage Handling Process at a Medium-Sized Regional Airport in Practice

Field research at a medium-sized regional airport in the Netherlands has provided insight in the baggage handling process at such airports operating in a point-to-point network. With regard to the baggage handling process at an airport operating in an international hub-and-spoke network (as depicted in Figure 5), several differences are identified.

At the regional sized airport studied, bags checked in at check-in counters and bags dropped at self-drop-off facilities are gathered at two conveyor belts which enter the baggage handling area. After entering, the first difference occurs in the security process of baggage items. Instead of five separate security screening levels, regional airports often apply fewer screening levels. The field research showed that the security screening has four levels:

1. Automatic x-ray screening. Since the number of bags handled at a regional sized airport is relatively small, there is only one x-ray machine per incoming conveyor belt.
2. Security staff can manually check the x-ray picture as made by the automatic x-ray screening machine in the first level and decide to accept or reject the bag
3. Security staff can run the bags that were rejected through a separate x-ray machine and examine the new picture
4. If the security staff feels there are suspicious items in the bag they can open the bag and if needed they can open the bag in front of the passenger to go through the items together with the passenger.

50 to 60% of all incoming baggage items continue to the sorting process after the first x-ray screening, where the x-ray machine automatically decides if the bag is free of suspicious items. This implies that 40 to 50% of the incoming bags proceed to the screening levels 2 and 3, where the first level x-ray picture is manually examined and a new x-ray picture can be made and manually examined by security staff. After this step, the majority of the bags proceeds to the sorting process. Roughly 2% of the bags continue to the fourth and final security level where the baggage items will be opened by security staff. Security staff can also decide to call the passenger and open the bag together so the passenger can give an explanation on the suspiciously looking items inside the bag.

In case passengers check-in or drop-off odd-sized baggage items, the odd-sized item needs to be placed on a different and separate, wider conveyor belt which enters the baggage handling area. At the end of this conveyor belt, a separate x-ray machine is located where security staff performs level 3 security screening and, if necessary, level 4. This means that odd-sized baggage skips the first two security screening levels and is always manually checked by either examining the x-ray picture from the separate x-ray machine or combined with opening the baggage item. After the item is cleared, it won't go into the sorting system but is put in a storage area from where it will be manually taken and placed on transportation trolleys when it is time to load the aircraft.

Another difference between the baggage handling as shown in Figure 5 and the similar system at a medium-sized regional airport is the presence of transfer baggage. At the second type of airport, the baggage handling system does not handle transfer baggage. This implies that the number of baggage items that enters the handling area after drop-off is the same as the number that leaves the area for transport to aircraft.

The most important difference between large hub airports and medium-sized regional airports is the elapse time of the various processes within the baggage handling area. In general, bags for continental flights can be checked in or dropped off between 3 hours and 40 minutes before departure. From 40 minutes before departure onwards bags cannot be checked in or dropped off. This implies that the process of baggage handling needs to be completed within 40 minutes. Within the regional airport studied, the total lead time – from entering the baggage handling area until placed on a baggage carts for transport to an aircraft – for non-odd-sized bags on average is six minutes in case the bag gets cleared after the first security screening level. This leaves plenty of time for the handler to transport the filled trolleys to the aircraft and place the bags into the aircraft. In case bags need to go to the second security screening level, one or two minutes need to be added to the lead time of six minutes. On average, it takes the handler 10 minutes to handle one bag from entering the area to the bag being placed on a baggage cart. In case level three screening is required, the lead time can increase greatly, but won't exceed the 40 minutes granted for the process. After the loading of the baggage carts, the bags leave the baggage handling area through one exit and head to the right aircraft. The time it takes to transport all bags from the handling area to inside the aircraft depends greatly on the position of the aircraft on the platform and the number of bags that need to be stacked inside the aircraft.

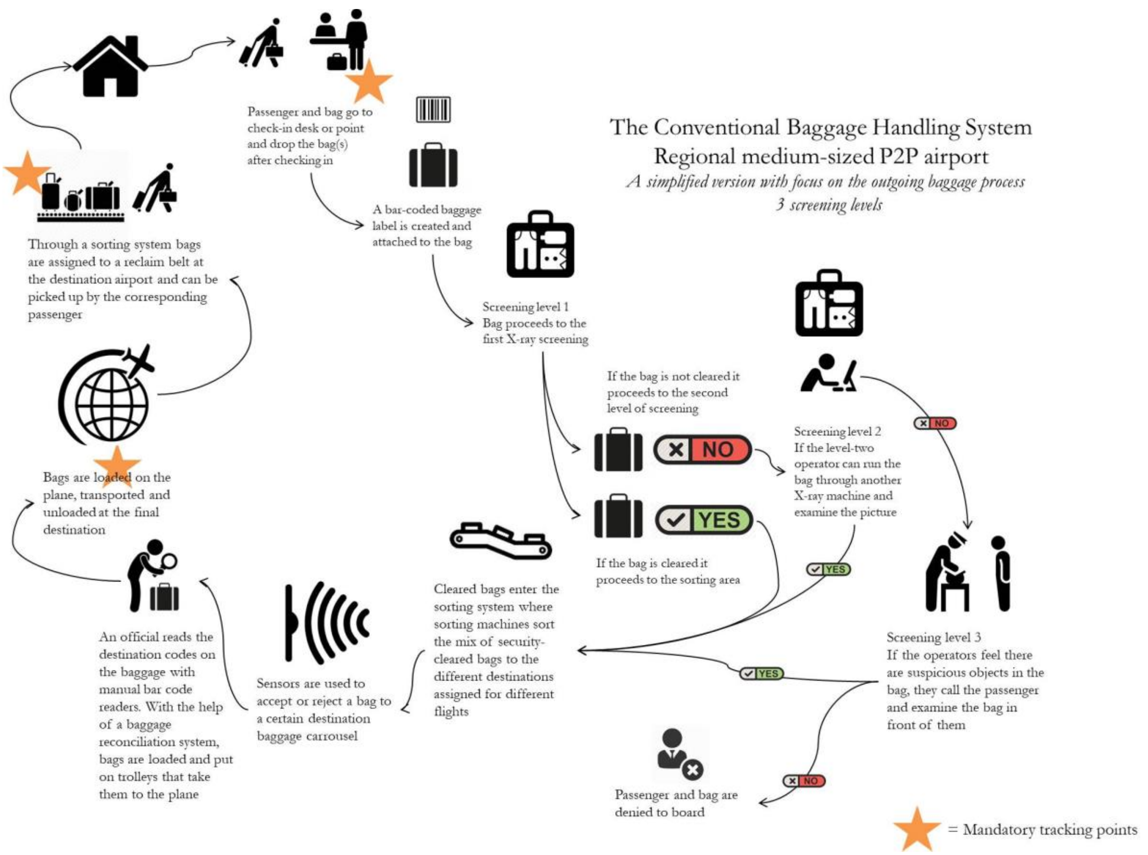

Figure 6 - Baggage Handling Process at a regional P2P airport (Image Author, 2017)

The opposite process – unloading bags from an aircraft and placing them on a reclaim belt at a destination airport – takes less time. At medium-sized regional airports the target time for this process is less than 15 minutes for loading the first bag on the reclaim belt. The last bag should be on the reclaim belt roughly ten minutes after the first bag, so all bags should be on the belt within 25 minutes after opening the aircraft doors. The baggage handling process at a medium-sized regional airport operating in a point-to-point (P2P) is shown in Figure 6.

The sorting, make up and transport processes can be executed by different handlers within the baggage handling area. Big hub airports like Schiphol Airport can have up to five handling parties operating in their baggage handling area. However, at the airport under study there is one handler that executes all these tasks. The security process is generally executed by a different company specialized in security.

2.1.2 Developments in Conventional Baggage Handling Systems

All around the industry, companies are working on the next generation of airport baggage handling. The focus mostly is on making airports 'greener', more efficient, more productive, safer and more cost-effective. Improvements in baggage handling system play a key role.

From automated sorters to integrated screening solutions for traceability, different parts of the conveyor belt based baggage handling system are being improved (Beumer Group, n.d.). But not only are elements added or automated in the system, the physical conveyors that are present in conventional baggage handling systems are also being optimised to make them more energy efficient (Clénet, 2010).

Baggage handling system component manufacturers continue to improve their existing products and develop new ones, like loading robots to automatically load bags on transport trolleys or unit load devices. This robot loading system has several benefits, the most important of which are efficiency improvements, cost reduction, health/ergonomics improvements and savings in space consumption. (Grenzebach, n.d.) claims that their loading system increases the throughput per baggage handler by a factor of 2.5 to 3.5 and delivers a reduction in operational costs per bag of 50% or more. For workers, the physical stress is reduced by using these loading robots, resulting in reduced indirect personnel cost. Furthermore (Grenzebach, n.d.) states that a 60 to 70% reduction in space occupation is possible.

These different developments contribute to more efficient or otherwise improved baggage handling systems. Airports that are most likely to benefit from these developments are larger airports, handling a large number of bags each day. The large volume of bags handled makes it easier to develop a sound financial business case for the required investment.

The next part of this chapter is about the use of automated guided vehicles (AGVs) in practice.

2.2 Transport Robot Systems

An example of a transport robot system that is used throughout many industrial branches is the automated guided vehicles, abbreviated to AGVs (Schulze, Behling, & Buhrs, 2008). Automated guided vehicles are defined as:

'An unmanned, self-propelled vehicle in the nature of a mobile robot has an on-board computer that stores path and machine function instructions and activates the drive and steering systems so as to cause the machine to follow a desired path.' (Field & Kasper, 1991)

This driverless transport system is used for moving materials horizontally from one location to another. The vehicles that perform the transportation job are part of the automated guided vehicle system – AGV system – which consists of multiple parts such as the vehicles itself, the transportation network on which these vehicles move, the physical interface between the production or storage system and the transportation system, and a central control system.

The vehicles are unmanned and move around the transportation network that connects al stationary installations in an area. The physical interface between the production or storage system and the transportation system are marked by pick-up and delivery points at workstations. These points mark the locations where load can be transferred to and from the vehicles. Depending on the type of vehicle, the load capacity differs. Load can be for example a container, a pallet or a single small package. Containers, pallets or other kinds of unit load devices (ULD) can hold multiple individual items. Transportation costs per individual item can be lowered by increasing the load. Another advantage of using ULDs is the lesser number of vehicles that are required in the system.

When loaded, the vehicles travel between pick-up and delivery points on paths (Vis, 2006). The desired path is usually indicated with targets in the area of operation combined with target readers on the vehicle itself, which together form a guidance system for keeping the vehicle on the predefined path (Schulze et al., 2008). The combination of all possible predefined paths shapes the layout of an AGV system. This layout determines the overall design of a system like a manufacturing system or a warehouse order picking system that make use of AGVs in their operation. The space utilization of these systems is determined by the layout, which in turn also affects the total distance travelled by the vehicles. Another component relevant in determining the possible paths and travel distances is the location of the pick-up and drop-off points for outward and inward bound items to be transported. By minimizing the total travel distance, decisions on the path design and the location of the load pick-up and drop-off points can be made (Goetz Jr. & Egbelu, 1990).

2.2.1 Control of Automated Guided Vehicles

To ensure efficient routing, scheduling and dispatching of vehicles, collision and deadlock avoidance, a high level of control is required. The activities that need to be performed by a controller are at least (Vis, 2006):

- Dispatching of loads to AGVs
- Selection of route
- Scheduling of AGVs
- Dispatching of AGVs to parking locations

The input to the automated guided vehicle system is transportation demand. If there is no transportation demand, vehicles are idle at a parking location where they stay until the controller assigns a new transportation demand to a vehicle. Once a new transportation demand occurs, a vehicle needs to be dispatched to handle the demand. The controller selects a vehicle and assigns a route and a schedule for the transportation task. One of the tasks of the controller is to make sure that the route and schedule assignment are such that the vehicle can execute the transport task without the occurrence of collisions and deadlocks. If the vehicle executed the transport task successfully and no new transport demand is present right away, the vehicle can be routed to a parking location where it awaits a new task to be assigned to the vehicle by the controller. Parking can also be combined with charging the vehicle.

The controller performs these tasks real-time due to the stochastic nature of the transportation process which requires control systems to be capable of making real-time decisions (Vis, 2006). This real-time control can be employed according to two control principles: decentralized control or centralized control. Central control refers to a single control system that simultaneously controls all AGVs in the system by allocating transportation jobs to the vehicles. The central computer assigns tasks to specific vehicles supported by a complete map of the area and the positions of all AGVs present in its memory, with which it continuously communicates.

Centralized control requires less physical links in terms of cables in the transportation area than decentralized control requires, which makes the layout easier to alternate or expand, but a more powerful computer is necessary to control the system as a whole. In the case of decentralized control, substations are present in the area, which are all connected to the central computer through continuous communication loops. Substations provide and control routing directions to the vehicles and control collision and deadlock avoidance.

The biggest advantage of decentralized control is the easier testing and troubleshooting. However, a disadvantage is that it is challenging to link all the substations and the central computer together (Vosniakos & Mamalis, 1990). Central controllers make decisions for the system as a whole and are often represented by discrete event systems. Decentralized systems on the other hand can be approached as agent-based systems, in which instead of a central controller making all the decisions for all vehicles in the system, the individual vehicles decide for themselves, leading to adaptive and optionally self-learning systems. Decentralized control is mainly suitable for large and complex systems in which many vehicles are present in a limited area, potentially leading to much interference between vehicles (Le-Anh & De Koster, 2006).

Next to real-time control, off-line control is also possible. Off-line control however requires perfectly predictable transportation requests and accurate information in the system. The origin, destination, release time and transportation time of transportation demands have to be known in advance so the controller can make decisions on dispatching, routing and schedule in advance to make off-line control possible (Vis, 2006).

Within the control of automated guided vehicle systems, decisions have to be made on different levels of the decision-making process in the design of such a system. The most important decisions in designing an AGV system are

- the design of the guide path
- the number of vehicles needed
- the scheduling of the vehicles
- the position where vehicles can stand idle
- battery management
- the routing of vehicles
- deadlock resolution.

The design of guide paths is considered a decision on the strategic level in the decision-making process since it influences decisions at other levels significantly. The tactical level consists of decisions on issues such as the number of vehicles needed in the system, the idle positions and battery-charging schemes. Decisions on the scheduling of the vehicles can belong to both the tactical and operational level. The operational level addresses decisions on vehicle routing and the prevention and resolution of deadlock situations. Decisions on different levels can influence each other like the design of the guide path has a direct impact on the number of vehicles that can or must be present in the system, which in turn influences the complexity of vehicle scheduling (Le-Anh & De Koster, 2006).

The design of guide paths on which automated guided vehicles rely is often discussed in literature (Gaskins & Tanchoco, 1987); (Goetz Jr. & Egbelu, 1990); (Majety & Wang, 1995); (Gourgand, Xiao-Chao Sun, & Tchernev, 1995); (Kook, Mook, Yoshimoto, & Hwan, 2002); (Wang & Chang, 2015). Most authors who addressed guide-path design in their work assume that the layout of the area in which the vehicles are operational is fixed, as well as the locations of the pick-up and delivery points. This simplifies the design of guide-paths to an optimization problem where the total distance travelled between pick-up and delivery points by the vehicles is to be minimized.

According to Le-Anh & De Koster (2006) guide-path systems can be classified by three characteristics; flow topology, number of parallel lanes and flow direction. Flow topology describes the complexity of the guide-path network. Table provides an overview of the different characteristics of guide-path systems.

Table 1 - Guide-path system characteristics

Flow Topology	
Conventional	Complicated network with paths, crosses, shortcuts and junctions Regularly used in warehouses and distribution centres
Single Loop	Simplest case, guide-path system consists of only one single loop Used in cross-dock centres
Tandem	Several loops grouped together, forming a tandem configuration Seen in manufacturing environments with grouped workstations
Number of Parallel Lanes	
Single Lane	A path segment in a network contains one lane
Multiple Lanes	A path segment in a network contains few parallel lanes
Flow Direction	
Unidirectional Flow	Vehicles can travel a lane in only one direction
Bidirectional Flow	Vehicles can travel a lane in both directions

In practice, the guide-path type is selected based on the characteristics of a facility and area and the experiences of the person responsible for the design combined with an expert system for support, since a clear guideline for guide-path system selection is lacking. After the type is chosen, a (mathematical) model can be used to design the most suitable guide-path system for the facility (Le-Anh & De Koster, 2006). In their work, Le-Anh & De Koster (2006) provide an in-depth overview on and comparison of guide-path design, stating that each guide-path system has its own advantages and disadvantages and is suitable for specific applications.

Estimating the number of vehicles needed in an automated guided vehicle system is an important decision on the tactical level of the decision-making process and influences the performance of AGV systems significantly (Le-Anh & De Koster, 2006). When an existing automated guided vehicle is chosen for the operation, factors as reliability and the range of operating speed can't be changed to affect the required number of vehicles. Factors that can be alternated and do affect the number of vehicles required are the guide-path layout, the location of load transfer points and the vehicle dispatching strategies (Egbelu, 1987).

What also influence the number of vehicles required is the capacity of the vehicles. Before calculating or estimating the required number, decisions should be made on the load one vehicle can transport. When vehicles with a bigger capacity in terms of load – multi-load capacity vehicles – are used instead of single-load capacity vehicles, the amount of vehicles needed in the system can be reduced (Le-Anh & De Koster, 2006). With several analytical or calculation-based estimation techniques, the number of vehicles required in the system can be determined. In his work Egbelu (1987) shows four methods to estimate the number of vehicles by using various information sources as input to the analytical models used. Information sources he used are for example the expected number of loaded trips between pick-up and delivery stations and the number of workstations present in the facility.

His four different methods lead to different results which are mainly caused by factors such as how empty trips are estimated and lost time caused by blocking. What Egbelu (1987) does not take into account in his methods are dispatching rules. Methods that do take this into account tend to provide a more accurate estimation on the number of vehicles required. Dispatching rules such as LIFO (Last In First Out) or FIFO (First In First Out) can be combined with other analytical approaches such as queuing models, multi-criteria decision modelling and network-flow modelling to improve the quality of the estimation (Le-Anh & De Koster, 2006).

When the design of an automated guided vehicle system is finished, the next step is to use it. Ever since their introduction in 1955 (Müller, 1983) the use of AGV systems has grown tremendously. Traditionally the AGV systems were most commonly used in manufacturing systems. Nowadays, AGVs are used for the internal and external transport of materials in a variety of applications and environments, such as warehouses and container terminals. The requirements for AGVs however differ per application. Transporting containers around container terminals requires a different AGV capacity than when an AGV is used to transport for example pallets or commercial packages in warehouses (Vis, 2006). The use of AGVs in both warehouses and container terminals will be elaborated on in the remainder of this chapter.

2.2.2 Automated Guided Vehicles in Warehouses

Especially in areas where repeating transportations patterns occur, AGVs pay off. Repeating transportation patterns can be found in for example industrial environments. An example of distribution areas can be found in the world of warehouses with cross docking centres (Vis, 2006). Warehouses connected to e-commerce operations frequently use AGV systems. E-commerce means that computers and telecommunication systems are used for conducting commercial transactions, which can be done 24 hours a day, 7 days a week (Ritter, 1992). As customers can place orders at any time during any day, e-commerce operations have to be flexible and need 24/7 operating hours. Having limited space needs, AGVs are a good fit for the distribution centres connected to e-commerce operations by eliminating the need for large spaces for facilities and reducing the labour intensity for workers significantly (Azadeh, de Koster, & Roy, 2017). Conventional warehouse operations require a large space since a lot of space is needed for storing items in racks, moving stock through the building, inspecting picked orders and allowing trucks to manoeuver and dock. Large e-commerce companies offer customers millions of unique items and have to deal with large and variable daily order volumes. In the process chain of delivering the right items to the right customers in time, order-picking by workers is the most laborious and expensive process. When this process is executed by humans, workers have to deal with a repetitive job with poor ergonomics and have to be willing to work in shifts which can also be during the night or in weekends. Finding the right people that are willing to do this job can be hard, which makes warehousing systems the ideal candidate for automation (Azadeh et al., 2017).

In their work, Baker & Halim (2007) refer to a paper of Rowley (2000) where he defines warehouse automation as:

"The direct control of handling equipment producing movement and storage of loads without the need for operators or drivers"

This definition covers not only the use of automated guided vehicles, but also includes equipment as automated storage and retrieval systems and sortation systems with conveyor belts. Automation in warehouses helps companies to improve their service and lower their costs, but the main motivation for most companies use automation in their warehouses is to accommodate future growth (Baker & Halim, 2007). Some warehouses are also combined with distribution centres. There are distribution centres that use AGVs to support their order picking process without eliminating the need for human workers. In these distribution centres or warehouses AGVs support the order picking process by bringing goods to the picker.

These so-called Pick Support AGVs (PS-AGVs) minimize the picker travel time to fill large orders (Azadeh et al., 2017). An application of a PS-AGV system can be found in the Netherlands at the fulfilment centre of PostNL in Houten. Their PS-AGV system called "AutoStore" can be classified as a GridSort system. The AutoStore system consists of AGVs that move over an aluminium framework while transporting plastic bins to operator stations. The top of the aluminium framework acts as rails in a grid on which the AGVs move but the rest of the framework can be used for the storage of plastic bins. The aluminium framework at the fulfilment centre of PostNL is 5.30 meters high and can store and transport 21.000 of these plastic bins. The 42 AGVs run on rails at the top of the framework with a maximum speed of 11 kilometres per hour, which limits them to forward, backward and sideways movements on the grid. One AGV can last 20 hours on a full battery and when fully empty, it takes charging stations four hours to fully recharge the AGV. It rarely occurs that an AGV runs completely out of battery power, since it returns to its charging station – there is one charging station for each individual AGV – after being idle for one minute. A benefit is that in this way the AGV is drip-fed power in the possibly short periods of time that it is at its charging station.

In warehouse operations, the Kiva Mobile Fulfilment System is a well-known example of the use of automated guided vehicles. In 2012, Amazon bought the Kiva Systems Company (Business Insider UK, 2017). This system uses identical transport robots in so-called pick-pack-and-ship warehouses. The Kiva robots are small enough to fit under 'inventory pods', which are three-foot-square shelving units. These pods consist of a stack of trays which are subdivided into bins in which goods can be placed. Figure 7 shows how the robots of the Kiva system lift the movable storage shelves to bring these shelves with goods to the worker.

Figure 7 - Kiva Mobile Fulfilment System (Amberber, 2014)

The overall design goal of the Kiva system is to keep the workers as busy as possible with the least amount of hardware, warehouse space and inventory. This requires good resource allocation algorithms. However, considering the resource allocation task as one big global optimization problem is impractical as resource allocation decisions have to be made real-time and the optimal solution also depends on the actual available paths and interactions of the vehicles, which is dynamic. Instead of considering global optimization, the allocation assignment problem is divided into several assignments:

1. Job assignment: assigning orders to workers at workstations
2. Pick-task assignment: once the job assignment is done, a robot drives to a pod, picks it up and drives to the worker
3. Replenishment-task assignment: goods in the bins of pods can run out. This assignment ensures in which bins in which pods the goods are replenished
4. Pod Storage: when a worker is done with a pod, the Kiva system selects an open position where the robot can park the pod

All movements of goods are done by the robots. This means that workers can stay at the same workstation while robots bring the shelving units to them. When a worker is done with an inventory pod, the robot stores it in an empty storage location. By moving the inventory to the worker instead of the other way around, the productivity of workers increases significantly (Wurman, D'Andrea, & Mountz, 2008).

2.2.3 Transport Robots in Container Terminals

As major seaports become increasingly accessible for deep-sea vessels, the popularity of containerization as a mode for maritime shipping and inland transportation rises. At the moment, multiple new deep-sea as well as hinterland automated container terminals are being designed worldwide. One of the choices in the design process regards the type of vehicle for container transport between seaside and landside. Two commonly used types of vehicles are Automated Lift Vehicles (ALVs) and Automated Guided Vehicles (AGVs), depicted in Figure 8 (Roy & De Koster, 2014). The performance of a transportation system within terminals that uses either of these two types of automated vehicles is studied by (Bae, Choe, Park, & Ryu, 2011). However, Roy & De Koster (2014) relax the term ALVs to AGVs. Both types are considered Automated Guided Vehicles that can transport only one container at a time, with ALVs having the extra property that they are able to self-lift containers.

Figure 8 – On the left side an ALV in a container terminal (SAE, 2008) and on the right side an AGV in a container terminal (Demag, n.d.)

In non-automated terminals, the transportation process between vessels and inland transportation is one of the least efficient and most costly processes. AGV systems can therefore provide benefits to both the port and its customers (Haefner & Bieschke, 1998).

AGVs are able to autonomously drive from a certain origin to a certain destination, but need an external device that loads and unloads containers from the AGV (Ottjes, Veeke, Duinkerken, Rijsenbrij, & Lodewijks, 2007). Therefore, in order to fit AGVs in terminal operations, quay crane and stack crane operations need to be synchronized with the AGVs. In Figure 9 a typical layout of a container terminal is depicted. AGVs can be used as a horizontal transport mode between quay crane operations and stack crane operations. In their model design of container terminal operations using AGVs, Roy & De Koster (2014) discuss this coupling between the vehicle and quay crane and stack crane operations, resulting in a queuing network model.

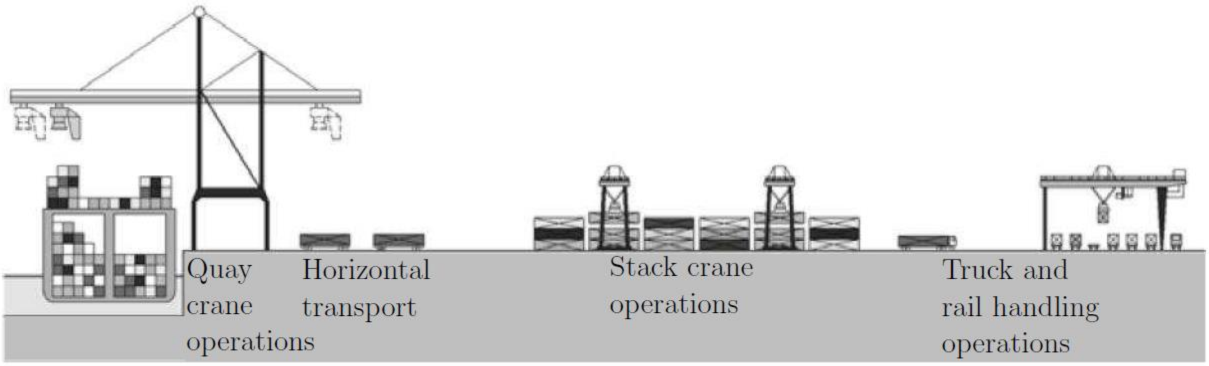

Figure 9 - Typical layout of a container terminal (adapted from (Roy & De Koster, 2014))

When AGVs are used in a container terminal, all vehicles use the same infrastructure. This makes the control of these vehicles essential for the performance of the terminal as a whole. Evers &

Koppers (1996) studied the horizontal transportation process at a container terminal with automated guided vehicles. They describe that although many forms and combinations of control exist – e.g. centralized control and decentralized control, both of which can be divided in zones – zone control is most frequently used. With zone control, the area in which AGVs operate is divided into zones. Each zone is individually controlled for movements within that zone however, central control is necessary to interface with other zones in the system. The general rule is that within one zone, only one vehicle can be present.

Evers & Koppers (1996) have introduced the 'semaphore' concept – abstracted from traffic lights – which controls the admission of vehicles into a zone, making sure the number of vehicles present in one zone does not exceed a specified maximum. As multiple transportation movements will take place simultaneously, many interactions can occur which can make zones critical. The authors argue that the complexity of these many interactions can be avoided by using distributed control concepts in which the transportation area is segmented in sub-areas or zones, combined with the concept of semaphore. It could happen that two or more vehicles want to enter the same zone at the same time. Such a situation calls for an 'access protocol' or dispatching rules to decide which vehicle can enter the zone first. A variety of dispatching rules is available. The theory of production control by Vollmann, Berry, & Whybark (1988) suggests roughly three dispatching rules that can be detailed and combined:

1. Random
2. First-In-First-Out
3. Priority

Priority can for example be given to a vehicle that travels in the same direction as the predecessor – which benefits the throughput – or to the vehicle with the earliest due time, or to vehicles having a direction with the smallest queue on the following semaphore. The difficulty with these rules in a decentralized control situation is that little to no information on the state of the total system is used by the semaphores. These rules can be considered to be local rules. The semaphores use the dispatching rules to decide whether or not to allow a vehicle into a zone. To make this decision, information may be used from sources such as information on the state of the semaphore itself, information on the state of adjacent semaphores, information on the type of vehicle, its direction and priority, and information form a supervising area controller. Even though the semaphore makes the admission decision on a decentral level it can still use information provided by a centralized area controller. The combination of using semaphores and dispatching rules in zone control results in a traffic control strategy that has been proven to decrease the amount of information needed to control AGVs in a container terminal and to increase the performance of the information system controlling the AGVs by the simplicity of the strategy.

2.3 Combining Baggage Handling Systems and Automated Guided Vehicles

Up to 2017, the developments in the world of baggage handling at airports have mainly focused on optimizing or improving components of the conventional systems. However, recently Vanderlande – one of the world's biggest material handling and logistics automation companies – has launched a new system called FLEET. This system combines baggage handling systems discussed in section 2.1 with transport robot systems as discussed in section 2.2.

FLEET uses vehicles that carry one bag at a time through the baggage handling area of an airport, as is visible from Figure 10. To route the vehicles through this area, FLEET is based on the technology of automated guided vehicles – AGVs. This system of AGVs eliminates the need for fixed conveyors and sorting system in a baggage handling area at an airport (Vanderlande, 2017). After a passenger drops his/her bag, barcode/RFID scanning takes place. The bag is transported into the baggage handling area via a small conveyor belt where it will land on one of the autonomous guided vehicles available. The vehicle brings the bag to one of the security screening facilities. As Figure 10**Figure 10** shows, the vehicle drops the bag on a small conveyor belt so the bag is transported through the x-ray by the conveyor belt and after the bag is scanned it lands on the same vehicle again. Depending on the wishes and needs and available space of an airport, multiple screening facilities can be used simultaneously, facilitating load balancing and a higher throughput. As the screening facilities are not parts of one big system of connected conveyor belts but rather are separate screening machines, expansion is easier as it is a matter of adding more screening machines in the baggage handling area and applying the same routing logic for the vehicles.

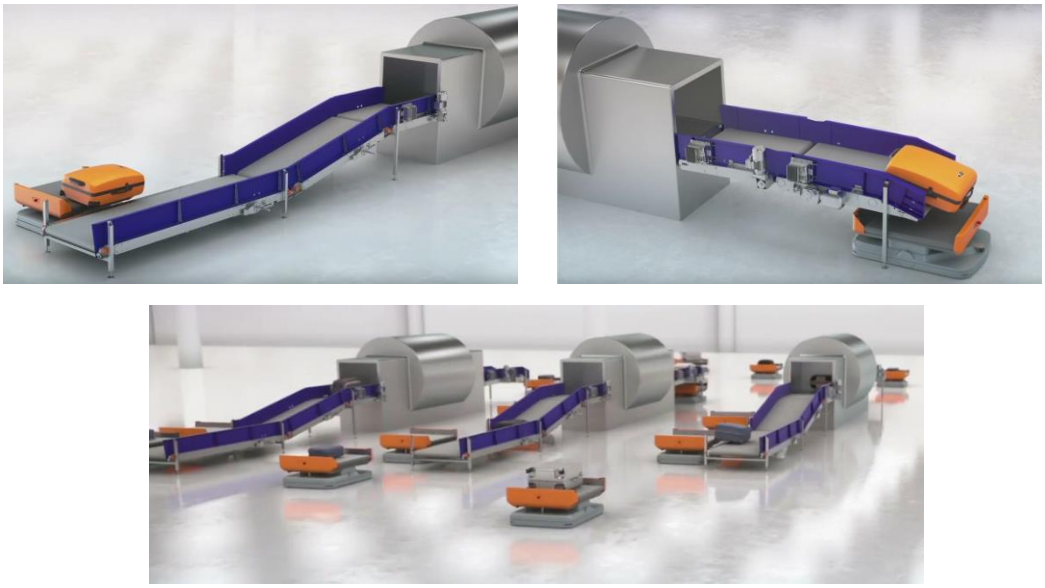

Figure 10 - Security screening with FLEET

If a vehicle is not used it can be parked at a central location where it simultaneously can recharge its battery. Having such a parking location makes the capacity of the FLEET system scalable. The sorting of bags is done automatically, as the robots are routed to the desired output location, bringing the bags to the correct makeup station right away. The vehicles transport the bags to one of the chutes leading to the different makeup stations. The bags are dropped off the vehicle and land on one of these chutes. The chutes end on a small circular conveyor belt at the makeup station for the destination the bag needs to go to. This process is unchanged compared to conventional baggage handling systems. This final sorting process is visible in Figure 11.

Figure 11 - FLEET's final sorting process

The use of these individual vehicles has several benefits such as flexibility and scalability, as it is easy to add or remove a number of vehicles from the baggage handling area depending on the capacity need. Next to that, the system is more sustainable than the traditional baggage handling systems with conveyor belts. Vanderlande claims that the vehicles used in their FLEET system require half of the energy of traditional systems and are designed to be recycled. Conventional baggage handling systems take up a significant amount of space in the baggage handling area. Transport robots are compact and can therefore handle more bags per cubic metre than the conventional system. The floor area saved can be used for other activities.

Next to capacity, sustainability and floor space benefits, an important benefit is resilience. In contrast to a conventional baggage handling system with conveyor belts – where a single bag that gets stuck can jam the entire conveyor belt and hence the entire baggage system – a system with individual transport robots is more resilient. In case a vehicle fails, other vehicles can bypass it and only one bag – the bag on the failing vehicle – is affected. This bypass possibility improves the operational continuity. The use of AGVs is efficient not only on an operational level, but it also relates costs to the actual bag volume handled (Vanderlande, 2017).

As can be concluded from this this chapter and Vanderlande's innovative FLEET system, there is a need for baggage handling systems that are not so dependent on spacious and laborious baggage handling systems that consist of many connected conveyor belts. The inflexibility of such systems can cause capacity problems when demand changes or develops differently than anticipated at the time of system design. Transport robot systems, particularly the widely used AGVs, have proven themselves as being suitable to achieve more flexibility in capacity and floor space needs in for example warehouses and container terminals. Vanderlande has combined the two systems into their FLEET system, using automated guided vehicle technology in a baggage handling area at airports.

2.4 Conclusion: Towards Autonomy in Baggage Handling

As mentioned, the FLEET system uses AGVs. The definition of Field & Kasper (1991) explicitly states that Automated Guided Vehicle systems follow path instructions. The on-board computer drives and steers the vehicle so it follows a desired path, following the path instructions. In these AGV systems this predefined path is considered free of obstacles so the vehicles can move from one point to another point following a desired path, without being interrupted. However, when an unexpected obstacle appears on the desired path, an AGV will stop in front of this obstacle. Other vehicles that follow the same desired path as the stopped vehicle will then encounter the stopped vehicle and will stop as well, resulting in a line of vehicles standing still behind the obstacle until the obstacle is removed. As paths are predefined, the line will continue to increase in length, which could result in an undesirable deadlock situation.

The FLEET system using AGVs is dependent on predefined paths that the transport robots follow. In case an unexpected obstacle appears on this predefined path the vehicles are not able to recalculate an alternative route themselves. A central system is required to assign the vehicle that has stopped in front of the obstacle to a different path.

To further explore the usefulness of individual transport robots, this research proposes a new concept where transport robots are used for baggage handling at airports. A concept like FLEET provides benefits when it comes to eliminating the need for fixed conveyors and sorting systems in a baggage handling area at an airport. This results in a more flexible and scalable baggage handling system, which is desirable to better cope with future changes in demand. However, opposed to the FLEET system that uses AGVs, this new concept uses transport robots that can autonomously determine the path they take to a makeup station, while being able to reroute real-time in case an unexpected obstacle – which can be another transport robot – appears to be blocking the route that was initially calculated.

Using these autonomous robots, the transport system is no longer dependent on grid-like structures present in AGV systems. This makes the transport system more adaptable and scalable than both AGV and conventional baggage handling systems. By eliminating the dependency on grid-like structures, this new concept can adjust to changes within an airport faster and easier. In the case of AGVs, the area where the transport robots will drive is mapped on this grid-like structure and paths are defined before the system is operational. When the area changes – for example when the baggage handling area is expanded or construction or maintenance work occupies part of the floor space in the area – all paths affected have to be redesigned centrally. With individual transport robots that are able to autonomously calculate their desired path real-time, such a central redesign of paths is not necessary. The vehicles themselves take the changed area into consideration when determining the proper path between two locations in the area.

Another advantage of autonomous transport robots compared to AGV or conventional baggage handling systems is the fault tolerance. Both AGV and conventional baggage handling systems have the weakness of having a low fault tolerance as both systems have a central unit, which implies a single point of failure. A single point of failure means that the entire system will come to a complete standstill when a single component or part of the system fails. In case the central unit of an AGV system fails, the whole system fails, making the system suffer from a lack of robustness (Khamis, Hussein, & Elmogy, 2014). A system with autonomous transport robots that (partly) uses decentralized control has a higher fault tolerance as it is not dependent on a central unit. The system as a whole is therefore less sensitive to the loss of this central unit. The absence of a central unit also eliminates the issues of scalability, flexibility and adaptability (Mauro, 2017).

Earlier practical and academic work discussed in this chapter provides an insight in how conventional baggage handling systems work and which benefits individual transport robots – AGVs in particular – can provide for systems with a transport task. By using individual transport robots in baggage handling areas of airports, a more adaptable, scalable and robust baggage handling system can be realized, especially when the individual transport robots are able to autonomously decide on preferred paths real-time. This research proposes a new concept of baggage handling with (autonomous) individual transport robots. This new concept is elaborated on in the next chapter.

3. The Baggage Robot Concept Description

This chapter concerns the use of autonomous transport robots in baggage handling system. In the previous chapter both baggage handling systems and transport robots systems were discussed. This chapter combines the two into a new concept: the baggage robot concept. First, a synthesis of the previous chapter tries to identify the usefulness of transport robots in baggage handling systems, followed by a section in which a rough idea is given on what this concept entails. Following from that, the most important elements in this concept are discussed in more detail. After the concept and the most important elements in it are made clear, requirements and constraints for this new baggage robot system follow as well as the key performance indicators that are defined to judge the performance of the baggage robot concept.

3.1 A Synthesis: Usefulness of Transport Robots in Baggage Handling Systems

The conventional baggage handling system at airports described in Chapter 2 uses a complex network of conveyor belts to transport bags from drop-off to the exit of the baggage handling area. Through this network of conveyor belts – sketched in Figure 12 – bags are transported through different levels of security and placed on a large sorter belt. Sorting machines connected to this large sorter belt sort the mix of security-cleared baggage to different makeup stations, corresponding to different locations. The result of this sorting process is that bags are sorted to a certain destination baggage carrousel at a makeup station, from where workers can load the bags from this carrousel onto baggage carts that are used to transport a batch of bags to an aircraft.

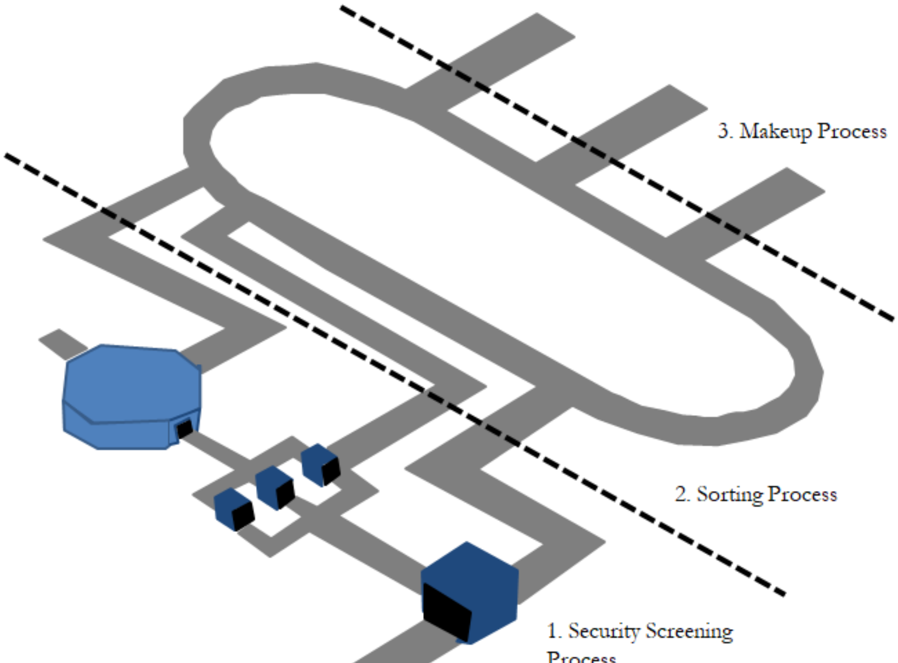

Figure 12- Conveyor belts in a conventional baggage handling system. (Image author, based on (Mehta, 2014))

Security screening machines are large and heavy and very hard to relocate once installed. This means that bags have to come to the security screening machines to be scanned instead of the other way around. An implication of these large and heavy machines being practically unmovable is that bags always have to travel to the same locations – where the machines are located - right after they are dropped off. Conveyor belts are considered the most convenient way to transport these bags from drop-off to the different security processes.

The second process the bags undergo is sorting. As can be seen from a rough sketch visible in Figure 12, a system of fixed and connected conveyor belts occupies a lot of space. The space in the middle can be considered wasted as it can't be reached from outside the conveyor belt network, unless the chutes to the makeup process are located in this middle area. This however requires the circular conveyor belt to be located higher than the makeup stations so the bags can leave the circular conveyor belt to be moved to a makeup station via a chute. Next to the sorting function, this central circular conveyor belt also functions as a buffer for early checked bags. However, this buffer is not of infinite size but limited to the size of this circular conveyor belt. Expanding this belt increases the capacity of the sorting system and buffer function, but this upscaling can't be done seamlessly or easily due to the rigid nature of conveyor belt systems and the need for a partial redesign of the routing rules. During the sorting process, bags are routed to one of the destination makeup stations where they reach a dead end in the conveyor belt system or the end of a chute. The use of conveyor belts or chutes to transport bags from the sorting process into the makeup area is considered to be non-problematic as bags reach a dead end there.

The disadvantage of using conveyor belts seems most present in the sorting process of the baggage handling chain. It is therefore assumed that the use of an alternative to a conveyor belt system will be most beneficial in the sorting process. This research is therefore focused on the sorting process within the baggage handling area at medium-sized regional airports, operating in a point-to-point network. The sorting process is an important part of the entire baggage handling chain and the location of this process is visualized in Figure 13.

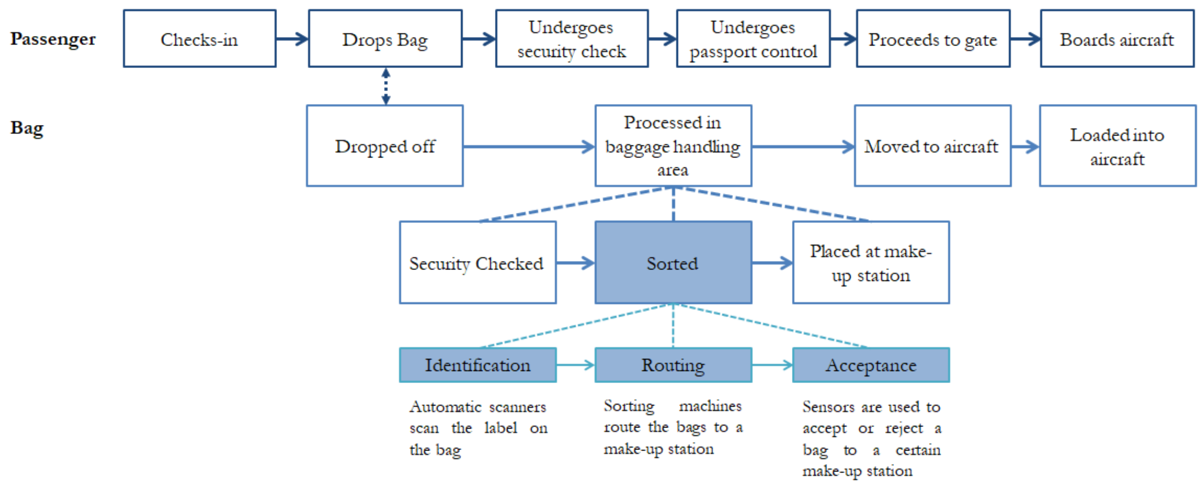

Figure 13 - Passenger and Bag Flows in the conventional baggage handling system

The Chapter 2 conclusion is that a more flexible baggage handling system can be realized by using individual transport robots in baggage handling areas of airports, especially when these transport robots are able to autonomously decide on their preferred paths real-time.

This chapter elaborates on the use of these new individual transport robots, specifically in the sorting process. As these robots are 'more intelligent' in finding paths through the area, they are assumed equally to more useful than AGVs in sorting processes. If the concept proves to be useful in the sorting process, use of these robots can be extended to other processes in baggage handling, including transportation of checked bags from drop off to security. This however requires future research.

3.2 The Baggage Robot Concept

The main purpose of a baggage handling system in an airport is to transport checked bags from bag drop off facilities to makeup stations, where the bags are loaded on baggage carts that bring them to the aircraft. The baggage robot concept serves the same purpose. By using individual transport robots that autonomously and real-time decide on their preferred paths, bags are transported between the bag drop facilities and makeup stations, replacing the complex system of conveyor belts that is currently used. Fixed machines such as the different security screening machines can stay at their original location and the transport robots are used to transport bags to, between and from these machines. A rough sketch visible in Figure 14 shows the difference between a conventional baggage handling system and the baggage robot concept. The conveyor belts in the security screening and sorting process are removed. The orange lines indicate the shortest paths between the security screening layers and three makeup stations. These orange lines do not indicate paths that the robots *have* to take, as robots can freely move in the baggage handling area. The lines rather indicate possible paths to provide the reader an understanding. The black and orange squares in the right sketch of the baggage robot concept indicate the individual transport robots moving in the area.

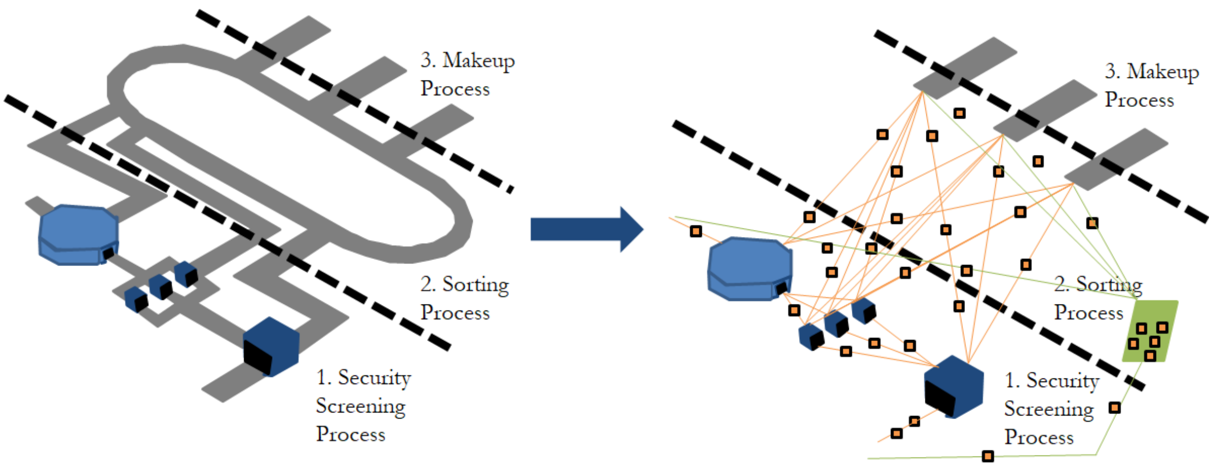

Figure 14 – Difference between conventional BHS and baggage robot concept

The robots can bring bags to and pick up bags from the security screening machines in a similar way as depicted in Figure 10, placing the bags on a small conveyor belt that goes through the security screening machine and pick it up after the screening is completed. The robots do not go through the machines. Once the bags are cleared in the security screening process they continue to the sorting process.

During the sorting process in a baggage handling system, bags need to be transported, from the area where bags are security checked to one of the makeup stations in the baggage handling facility. By using individual transport robots instead of conveyor belts, bags can be transported to the correct makeup station directly. By taking the bag straight to the makeup station it needs be delivered to, the robots fulfil the transport and sorting task simultaneously.

As each individual robot has a battery, a charging facility is needed to charge these batteries. The battery charging is done by inductive charging. Inductive charging positions can be anywhere in the area. The sketch indicates a charging area (in green) with multiple charging positions. The green lines show paths that robots *can* but don't *have* to take to these charging positions after having delivered the bag they were carrying at one of the makeup stations. The charging area serves a double function as it can also function as storage area. The same drip-feeding technology of AGVs can be applied here. Robots can be drip-fed while standing idle in the charging and storing area, making sure the battery is continuously being charged during idleness. Robots that have completed a transport job, i.e. have picked up a bag from a bag drop facility and unloaded the bag at the correct makeup station, can take on a new transport job as long as their battery is sufficiently charged to complete another transport job.

In order to execute the sorting and transportation task described above, elements such as the layout of the system, the way in which robots find their way through the baggage handling area without colliding into other robots and the way in which this system as a whole is controlled need to be made explicit. The next section describes the most important elements in this baggage robot concept.

3.3 Elements of the Baggage Robot concept

The previous section concerned the baggage robot concept in general. This section focuses on the most important elements in this baggage robot system. It is important that these elements are designed in such a way that the system as a whole can perform its transportation and sorting task. These elements are layout configuration and charging, the control of the system and the routing of the transport robots. Two elements that are included and are linked to routing are collision avoidance and deadlock resolution.

As the baggage robot concept is a system with multiple robots that operate in the same environment, it's considered a multi-robot system (MRS). In the last decade multi-robot systems have become an important area of research in robotics due to the challenging nature and many potential applications (U. Lima & M. Custódio, 2005). The existing knowledge on these multi-robot systems are used to design the elements discussed in this section.

3.3.1 Layout Configuration and Charging

The layout of a baggage handling area in which individual transport robots transport bags from the drop off point to a makeup station can be adjusted easily. The robots are not dependent on a lot of fixed infrastructure. The only element they need in order to remain operational is a place where they can charge their battery. As mentioned in section 3.2, charging is done by inductive charging. Inductive charging uses an electromagnetic field to transfer energy and therefore does not require any cables. By integrating electromagnetic induction in the floor of the baggage handling area, the charging infrastructure does not result in any obstacles for transport robots in motion. The location of charging positions can differ per baggage handling area, depending on the original layout of the area when a baggage robot system is integrated in an existing baggage handling area. Alternatively, it can be decided upon during the design phase of a new baggage handling area.

Figure 15 - Inductive Charging in Fleet

Vanderlande's FLEET system also uses inductive charging. Figure 15 shows how FLEET AGVs charge. The charging area also functions as a storage area for robots. As the image shows, the charging infrastructure itself does not cause any obstacles, but robots charging themselves can form an obstacle to other robots. By keeping this in mind, locating the combined storage and charging area on the sides of the baggage handling area causes the least inconvenience for robots in motion, driving from one end of the baggage handling area to the other end.

As the baggage handling area is not open 24 hours a day as at a regular medium-sized regional airport check-in and bag drop is only possible in a specific time slot that can change from country to country and from airport to airport. In the Netherlands for example, passengers can drop their bags from 05:00 to approximately 22:20 due to flight restrictions during the night. This means that in this specific case, transport robots don't have to be operational between 22:20 and 05:00. The robots can use these nightly hours to fully charge themselves. The first batch of incoming bags in the morning can be handled by fully charged robots. During operational use their battery level decreases. When this level drops below a set threshold, the robot needs to proceed to a charging position and charge until the battery level at least exceeds the threshold.

Depending on the number of robots in the system, the number and the arrival pattern of bags that enter the baggage handling system, charging strategies might be of importance. When the number of bags that enters the baggage handling system in the time frame baggage drop-off is allowed equals the number of robots in the system, each robot only has to transport one bag a day. As all robots start the day with a fully charged battery and transporting one bag does not consume 100% of the battery level, additional charging strategies are not necessary. However when there are fewer robots than incoming bags, some or all robots have to perform two or more transport tasks a day. Depending on the arrival pattern of bags, additional charging strategies might be relevant. For example at an airport that has only three flights a day and these flights are scheduled in such a way that the bag drop time frames do not overlap and the time between these flights is enough for a robot to fully charge its battery, the number of robots necessary can be reduced to the maximum number needed to handle the largest flight. Charging strategies are also dependent on technical specifications of the transport robots such as charging rates and battery consumption rates. All factors influencing charging strategies can differ by airport and should therefore be determined case by case.

3.3.2 Control

One of the most important considerations in the design of a new transport robot system is the control architecture, in particular the technique used to coordinate the motions of the individual vehicles (Mas & Kitts, 2010). In multi-robot systems, the design of the overall control architecture for the individual transport robots influences the robustness and scalability of the system (Parker, 2009). Parker (2009) distinguishes the four most commonly researched architectures for multi-robot systems, being:

1. centralized architectures
2. hierarchical architectures
3. decentralized architectures
4. hybrid architectures

(1) Centralized architectures use one central unit that coordinates all individual robots. However, such control architecture is rarely used in practice as having one central unit is risky in terms of robustness; there is a single point of failure in this architecture. Next to this vulnerability, real-time communication is considered challenging. In a centralized architecture, all individual robots need to communicate their state to the central unit, which needs to translate these individual states into a central state and act accordingly. This communication loop between all individual robots and the central unit however takes time and a high frequency of this communication is necessary to realize real-time control. This makes real-time control in centralized architectures difficult. Centralized control architectures are found to be best suitable for systems in which the central control unit is able to oversee all the individual robots in the system and send the same instructions to all these robots simultaneously (Parker, 2009). An important condition here is that the individual robots in the system strictly follow the instructions of the central control unit and are not allowed to deviate.

(2) In hierarchical architectures the control perspective is that of an individual robot. Each robot keeps an eye on the states and actions of a small group of other robots. This small group oversees another, smaller, group and so on until there are no groups left to oversee but one individual robot. This individual robot does not keep an eye on anyone else as everyone else in the system is already overseen. This left over robot simply has to do its own job, without minding

other robots. All other robots have control over other robots in the system. This type of architecture is easier to scale than centralized architectures and does not have a single point of failure. However it does have a disadvantage, being the recovery of failures of robots high in the control hierarchy (Parker, 2009).

(3) Decentralized control architectures do not depend on one central control unit. Instead, each individual robot takes actions based only on knowledge they get or retrieve from their direct and local environment. Unlike in centralized and hierarchical architectures, the individual robots in a decentralized control architecture are not controlling anyone but themselves, making them responsible for themselves only. The downside to this individualistic perspective is that global knowledge of the system is missing (Parker, 2009). The incorporation of high-level goals for the system as a whole is therefore problematic, as these goals have to be incorporated into the local control of each individual robot. This may lead to optimal local solutions that deliver sub-optimal global outcomes. Decentralized control architectures however are the most commonly used control architectures in multi-robot systems.

(4) Hybrid control architectures combine the local control of decentralized control architectures with higher-level control approaches. This architecture benefits from the robustness provided by the decentralized control architecture that eliminates the single point of failure and from the ability to influence the actions of all individual robots that centralized control architectures provide. Many multi-robot systems use of hybrid control architectures (Parker, 2009).

Robots in the baggage robot concept are considered to be autonomous. They have some control over their state and behaviour and are able to react to actions of robots in their proximity. A decentralized control architecture therefore seems most suitable as this architecture provides the individual robots with decision authority. However, to obtain a system that is capable of avoiding collisions and resolving deadlocks when they occur, some centralized control aspects need to be incorporated to improve the performance of the system as a whole. The hybrid control architecture provides a combination of decentralized and centralized control, making it suitable for the baggage robot concept as it increases the robustness, scalability, flexibility and performance of the system.

3.3.3 Routing

As this research focuses on the sorting process specifically, it is assumed that all bags that arrive at the sorting area are cleared in the security process. In order for bags to arrive at a makeup station that corresponds to the destination the bag needs to end up at, individual and autonomous transport robots have to transport the bags through the sorting area. For passengers it is important that their bags are loaded into the same aircraft as themselves and in time. To create as much slack time as possible in the sorting and other processes of baggage handling, the preferred path bags travel between the entrance and exit of the sorting area is the shortest path. Figure 16 shows the shortest and direct paths or routes that can be used to transport bags in a hypothetical sorting area with *m* entrances and *n* exits.

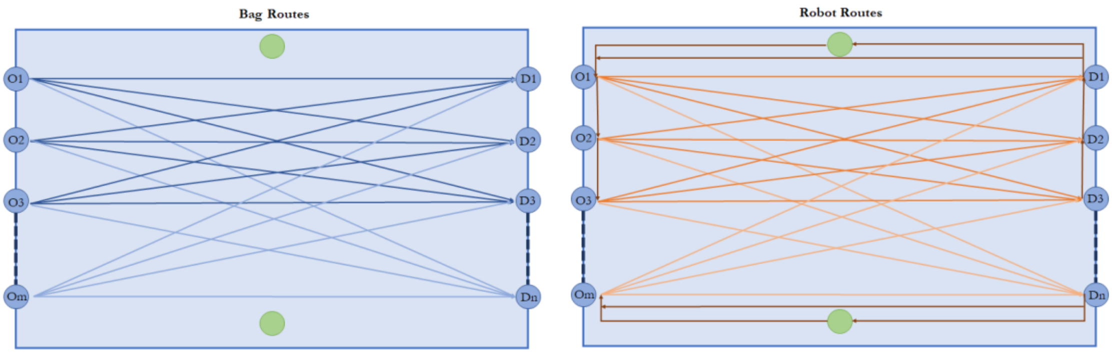

Figure 16 - Bag routes (left) and robot routes (right)

As bags cannot move themselves, they depend on robots for their transport. As robots do not exit the sorting system when they unload a bag at a makeup station, they also need paths or a route back to the entrance to pick up new bags. These return trips can be interrupted when the robot runs low on battery power. In that case, the robot needs to go to one of the storage and charging area positions (marked with a green circle in Figure 16) until it is sufficiently charged and can continue to one of the *m* entrances. The location of the storage and charging positions or areas depends on the chosen layout; Figure 16 only serves as an example. When there are no incoming bags, robots can use this same storage and charging area to stand idle and drip-feed their battery, until bags start coming in again. The paths or routes a robot can take therefore differ from the paths or routes bags preferably take in the sorting system and are visible on the right in Figure 16. Robots can transport bags from the entrance to the exit of the sorting area by taking the shortest path. These trips, where robots are transporting bags, are more important timewise than the empty return trips.

3.3.4 Collision avoidance

When a number of robots have to perform their transport tasks in the same area, as is the case in this baggage robot concept, a risk exists that they collide while moving around. To avoid this risk, two basic approaches are possible. The first one relies on centralized control while the second approach is hybrid.

The first collision avoidance approach is only applicable for systems that require robots to follow a predefined and fixed path like AGV-systems. In such systems robots are not free to decide their own path nor can they change the path that has been predefined for them. The central unit in such systems can determine these fixed paths for all robots simultaneously in such a way that the paths for the robots do not cross at any point. These so-called collision free paths eliminate the risk of collisions altogether, but can be inconvenient or impossible for large or complex systems (Jäger & Nebel, 2001). Another way to avoid collisions in systems with centralized control is the addition of motion controllers in the system that assign initial delays to specific robots, making sure that no two or more robots encounter each other at the same location (Zhou, Hu, Liu, & Ding, 2017). These two versions of the collision avoidance approach are applicable in systems with centralized control. However, as mentioned in section 3.3.2., centralized control approaches are not preferred in the baggage robot concept. Using centralized control for collision avoidance requires a lot of computational power and a global communication network to communicate with all robots in the system.

The second collision avoidance approach can be applied to systems where robots have flexible paths and are able to change these paths at any time, like the baggage robot system. The robots plan their paths independently. The path they initially plan is collision free, but there is no guarantee that this path will remain collision free as all robots in the system are able to change their own paths at any time. This may result in paths crossing each other at a certain point in time. In such a case, it can happen that robots have already started moving along their path when they detect another robot getting close to crossing their path (Zhou et al., 2017). The detection of other robots in their proximity is one of the most important abilities of the robots to realize collision avoidance. The most effective way to detect other robots close by is by obtaining information on the planned paths of these closest robots by means of local communication (Arai, 1999). To avoid a collision between the robots, re-planning of one of these paths is necessary to guarantee collision free paths for both robots involved (Zhou et al., 2017). To decide which robot needs to alter its course, a centralized component can be used to coordinate this decision, making sure that no collisions will occur.

This form of hybrid control for collision avoidance uses centralized components to achieve global coordination by decentralized algorithms and assumes only local communication between pairs of physically close robots. The local communication between the robots allows for more adaptive coordination between the robots when planning their paths, supported by a centralized component (Jäger & Nebel, 2001).

As has been established in section 3.3.2., the baggage robot concept uses a hybrid control architecture. In this architecture, robots can communicate with other robots in their direct surroundings, while moving around in the baggage handling area. In this concept, the robots move towards their goal – either a bag at an incoming conveyor belt or a makeup station to drop the bag or a charging and storage position to stand idle and/or charge – at their maximum speed when no objects or other robots block their paths. However, a robot can detect an obstacle – which can also be another robot – while moving. When a robot approaches another robot and the distance between the robots decreases, it needs to change its current behaviour to avoid colliding with the other robot (Arai, 1999).

When two robots approach each other and the distance between them decreases, they exchange information on their positions and planned paths. By exchanging this information they detect if and if so where they may run into each other on their planned paths. To avoid a collision, the robots need to communicate and coordinate their movements (Jäger & Nebel, 2001). Together the robots need to decide who gets permission to go and who has to give way to the permitted robot. To make this decision, predetermined traffic rules can be used. These rules depend on the type of predicted collision.

In his research on multi-robot systems with autonomous robots in a parcel sorting system, Mauro (2017) distinguishes two types of collisions for multi-robot systems: side collisions and frontal collisions. The traffic rules for these types differ and are also suitable for the baggage robot concept due to the similarities in robot type between his and this research. In case of an imminent side collision, stopping and resuming policies are used to avoid actual side collisions. In case a side collision is imminent between a robot that is transporting a bag and a robot that is empty, the robot that is transporting a bag gets priority over the empty robot. The empty robot needs to stop moving until the loaded robot has passed. The stopped robot can resume its path once the loaded robot has passed completely. When two robots with the same priority are threatened by a side collision, priority cannot be decisive and one of the robots will be randomly picked to move first. This random picking can be done by the centralized component or the two robots negotiate who goes first by for example "pulling straws". In case of a possible frontal collision, Mauro (2017) used a right moving policy to be executed by both robots, regardless of the priority. Both robots stop moving forward and instead move one step to the right and then re-calculate and resume their new shortest path. Figure 18 shows the movements of the robots in case of an imminent frontal collision. The grey arrow in step 3 and 4 indicate the path of the robot that have to wait. This research proposes a more relaxed version of the right moving policy, being a turn, (wait) and continue policy. Both robots stop moving forward and instead move one step to a neighbouring position that is unoccupied and then re-calculate and resume their new shortest path. If all neighbouring positions are occupied, robots wait until one of these neighbouring positions is vacated again, after which it will resume a shortest path. A right-moving policy is considered but found to be too constraining as when the position on the right is occupied, deadlock is lurking.

To determine when a collision avoidance measure should be invoked, the baggage robot concept uses a safety zone for each robot. This safety zone can also be referred to as a safety distance d_{safe} between two robots. When the distance between two robots is larger than or equal to the safety distance, no possible collision will occur in the near future. When this distance becomes smaller and the safety zones start to (partially) overlap, one of the collision avoidance measures is invoked to avoid an actual collision, depending on whether the possible imminent collision is a side collision or a frontal collision. Figure 17 shows an example of the right moving policy to avoid a frontal collision and Figure 18 shows a stopping and resuming policy where the robot coming from south and heading to the north gives priority to the robot moving from west to east. In both figures, red marked areas indicate the areas where the safety zones of two robots overlap each other. These overlapping areas trigger the collision avoidance measures.

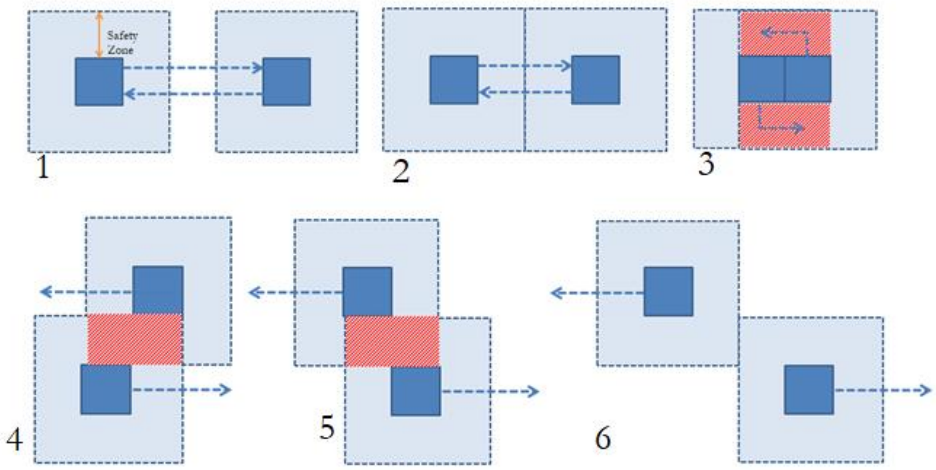

Figure 17 - Frontal collision avoidance, (Image Author, 2017)

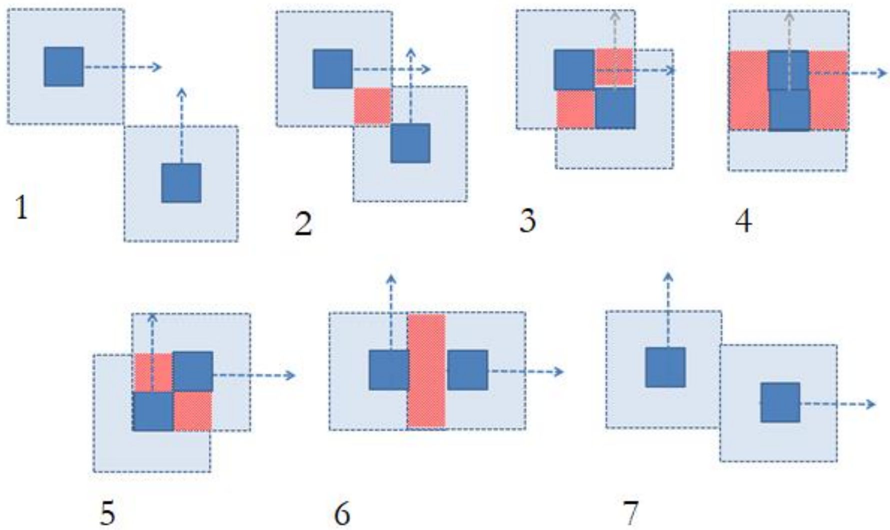

Figure 18 - Side collision avoidance, grey dotted line indicating the path of the robot that has to wait (Image Author, 2017)

3.3.5 Deadlock resolution

When robots are able to plan their shortest paths and are able to react to the movements of other robots in their proximity, a system with robots that move over paths that are as short as possible without colliding into each other is realized. However, when trying to design such a system, it can hit a situation where two or more robots get stuck in an impasse, as they are waiting for one another to take action. As time continues, this can even lead to a total deadlock situation in which all robots are waiting for each other and where no robot is able to change its waiting and indecisive state. Such a deadlock situation is considered undesirable, especially when it lasts a long time, leading to a situation where bags are not delivered to their correct makeup station in time, resulting in bags being mishandled as they can't be loaded into the aircraft in time.

Systems that have a fully decentralized control approach are dependent on local communication between robots as a central unit is missing. Systems that have a fully decentralized control approach are dependent on local communication between robots as a central unit is missing. As local communication can lead to a deadlock situation in which robots end up in a cycle, some form of external intervention is needed to resolve the deadlock. This means a fully decentralized approach is not a suitable approach when deadlock resolution is desired.

Deadlocks can occur in many different ways. Usually when a situation occurs in which particular robots end up in a cycle where they can't escape from without external intervention. These situations can be caused by certain system conditions. Examples of conditions in which deadlocks are hard or impossible to resolve are:

- If none of the robots has an unaffected free space in its direct neighbourhood where it can go to, making it impossible for a robot to plan a new free path as long as all his neighbouring positions are occupied and stay occupied
- When there are too many robots in an area that is too small for all these robots to plan a free path and ensure some leeway to reroute
- When a lot of robots have to go to the same point at roughly the same time. In case of a deadlock cycle between two robots, the deadlock situation worsens over time quickly.

The more robots are involved in a deadlock, the harder it gets to resolve the deadlock. That is why it is preferred to prevent a deadlock from occurring altogether. The earlier discussed centrally defined 'traffic rules' can help in avoiding deadlocks from occurring. By invoking the collision avoidance measures earlier discussed when collisions are imminent, deadlocks are avoided too. As the turn and continue policy is a relaxed version of the right turn policy, robots have more neighbouring positions to go to. By adding the possibility to wait one or more time steps in this policy, complete and insolvable deadlocks are avoided.

3.4 New System Requirements, Constraints and KPIs

Before designing a sorting system for baggage handling systems, where the sorting will be executed by autonomous and individual transport robots, system requirements need to be established. There are two types of requirements to be distinguished; functional and non-functional requirements. Functional requirements describe the functions that the physical system should have in order to function properly. Non-functional requirements on the other hand, help to assess the quality of the operational system. Next to meeting these two types of system requirements, a system that is newly designed should comply with certain constraints. Constraints refer to the limitations on these requirements and the conditions under which the sorting system is expected to function. The system requirements and constraints concern the design of the new sorting system for baggage handling systems but to measure the performance of this new system, key performance indicators have to be established. Key performance indicators measure the performance of the system and by doing so, can show gaps between current and desired performance (Weber & Thomas, 2005). This section deals with the system requirements, constraints and KPIs relevant for a sorting system in a baggage handling system, where the sorting is executed by individual and autonomous transport robots.

3.4.1. Functional Requirements

The basic functionalities of a sorting system in a baggage handling context at airports are that the system needs to *transport bags* from the entrance to the exit and that the system needs to *sort the bags* correctly. Within the scope of this research, the first basic functionality entails that the system should be able to pick-up bags that are cleared in the previous process, and transport them to the start of the next process, the makeup, where bags are loaded into baggage carts that will be transported to the aircraft. The second basic functionality is about the sorting. As a bag should be loaded into the same aircraft as the passenger it's associated to, it needs to be transported to a specific makeup station. As passengers can drop-off their bags at any drop-off point that is operational, the sorting system should be able to identify to which particular makeup station it should sort the bag to.

A passenger that likes to check-in a bag of a certain *size* and *weight* can drop this bag at one of the airport's drop-off facilities. At the drop-off facility, a baggage label is attached to the bag, indicating amongst other information, the destination airport that the bag needs to be transported to. Once the bag is dropped off, it enters the sorting area via a small conveyor belt. Upon entering the sorting area, the baggage label is scanned by *scanners* that identify the destination of the bag. After the label is scanned the bag waits for the first available transport robot. Once a transport robot reaches the rear end of the drop-off conveyor belts, the robots needs to *communicate* to this conveyor belt that it is ready to receive the bag. Once a bag is loaded on one of the transport robots, the robot starts *transporting* the bag with a certain *speed*, while making sure the bag is *supported* well to prevent it from falling from the robot while the robot is in motion. The transport robot *sorts* the bag it carries based on the information acquired by the system of scanners, so the bag is transported to the correct destination makeup station. While in motion, the transport robot uses its battery power. Once it reaches a certain battery level threshold it needs to proceed to a charging area, where the battery is recharged by *inductive charging*. If a robot is not needed to perform a transport task, it can remain in or move to a combined storing and charging area where it can recharge or stand idle if the battery power level is above a threshold. Once the transport robot is idle for a certain amount of time, it can *shut down* to save battery power. If the robot is needed again, the system of scanners triggers the robot to *activate* itself after the scanners scanned the baggage label of an incoming bag and the robot should start a new transportation task.

The three main tasks of the autonomous and individual transport robots in this sorting system are (1) picking up bags from the drop-off stations to start the sorting process, (2) sorting while transporting bags to a chute ending at a correct destination makeup station and (3) dropping bags onto chutes that end at a makeup station.

The functional requirements mentioned above are summarized in Table 2. The requirements are divided in roughly three sections: the sorting system as a whole, the specific task of transporting bags and the requirements for the system when robots are not transporting bags.

Table 2 - Functional Requirements

	Functional Requirements

Sorting System
 	
Transporting Bags
	• Communicate with scanners at the entrance belt to trigger robot activation and to obtain destination information of a bag
	• Pick-up bags by communicating with the drop-off conveyor belts to signal the conveyor belt that it can transfer the bag onto a transport robot standing ready
	• Transport bags
	  – Handle bags of a certain size and weight
	  – Provide support to prevent bags from falling off a transport robot
	• Sort bags
	• Drop off bags at chutes ending at a makeup station

While not transporting bags	• Temporarily deactivate a robot when it is idle for a certain amount of time • Send robots to a charging area when the battery power level of a robot is under a threshold level • Transmit inductive power from charging areas to transport robots

As in this research bags are simplified to standard bags, so no odd-size or heavy bags, the sorting system considered handles bags that don't exceed 32 kilos in weight and have maximum dimensions of 100 cm in length, 65 cm in height and 75 cm in width. All types of baggage that do not match these weight and size restrictions are considered odd-sized and out of scope.

3.4.2. Non-functional requirements

Next to functional requirements, non-functional requirements can be identified for the sorting system that uses autonomous and individual transport robots. First and foremost, the system should provide *capacity flexibility* to the baggage handling process. The main purpose of replacing the static conveyor belts by dynamic transport robots in the sorting system is to make the capacity of the sorting system more flexible so the system can adjust to (seasonal) variations in demand. Next to adjusting to temporary changes in demand, the system should also provide *flexibility in expanding* when demand is expected to change on the long term. While operational, the system should not be over dimensioned, meaning the system is larger than necessary, resulting in a low *utilization rate*. This goes together with the *capacity* that the system is able to handle given a specific configuration in terms of the number of bags sorted and transported per time unit. The capacity and utilization rate can be influenced by *deficiencies* in the system that should be between bounds to be acceptable. The system should be *resilient* when deficiencies occur to maintain sufficient uptime for the system to be considered *reliable*. In case a deficiency does occur, or preventive *maintenance* is needed to prevent severe deficiencies, it should be doable to perform maintenance on the system. If human intervention is needed to resolve a deficiency in the operation of the system, it should be *safe* for workers to enter the transport and sorting area and do their work. The non-functional requirements stated are summarized in Table 3.

Table 3 - Non-functional requirements

	Non-functional Requirements
Sorting System	• Capacity flexibility • Expansion flexibility • Utilization rate • Capacity • Deficiency resiliency • Reliability • Maintainability • Safety

In order for the new system to be feasible, it should comply with these non-functional requirements in at least the same way as the traditional sorting part of the baggage handling system does. Preferably, the new system should comply better with these non-functional requirements, meaning for example the capacity should be at least the same but preferably higher compared to the traditional sorting system and the system should be equally safe in order to be a suitable replacement for the traditional system. Non-functional requirements like investment costs, operational costs and reusability are not within the scope of this research, but are important to consider when deciding on researching the concept further with the purpose of implementing it in an airport baggage handling environment. This research focuses on the first five non-functional requirements. These non-functional requirements – except for expansion flexibility – are roughly translated into KPIs, which will be elaborated on in section 3.4.4. The expansion flexibility is in the nature of the autonomous and individual transport robot system when using a hybrid control approach. The non-functional requirements reliability, maintainability and safety are not included in the continuation of this research, but are mentioned to give a more complete insight in the non-functional requirements that the baggage robot concept as a whole should comply with and not only the sorting process part of the concept.

3.4.3. Constraints and Basic Conditions

Constraints refer to the limitations on the functional and non-functional requirements and the conditions under which the sorting system is expected to function. It all starts when bags enter the sorting area through a maximum number of drop-off facilities, representing the *input capacity* of the system. The bags that enter have to comply with previously stated maximum *weight and size* restrictions in order to be accepted into the sorting system. When bags enter the sorting area, individual and autonomous transport robots pick up the entering bags. These *robots* also have a *capacity*; they can only carry one bag at a time. Depending on the number of robots present in the sorting system, the *sorting system capacity* is determined. This capacity reaches its maximum in a situation where all robots are fully charged and operational at the same time. As one robot can transport one bag, the capacity of the sorting system equals the number of operational robots present in the system. In order for a robot to transport a bag from the entrance of the sorting system to the exit where bags are exiting the sorting system and entering the makeup area via a chute, it needs to move with a certain *speed*. As autonomous and individual transport robots are currently not yet operational, the maximum speed is based on existing practice of a similar system which is operational in a sorting system of a fulfilment centre in the Netherlands. As the goods these existing robots transport generally have a significantly smaller weight, a margin is taken for the speed of the robots in the new baggage sorting system. Heavy bags up to 32 kilogram each need to be transported at the same speed as lighter bags, so the speed is set to 7 kilometres per hour. The acceleration and deceleration rates of the transport robots are not included in this research as they are considered too detailed for the purpose of this research. When moving with this constant speed in the sorting system, the time it takes to reach one of the *exits* representing destinations at the other end of the area also depends on the *size and shape of the sorting area*. For this research, a big open square or rectangular space is considered without obstacles on the floor so the robots can move freely from one end to the other end of the area. Different floor shapes influence the routes robots can take as well as the risk of collisions between robots. The constraints mentioned above are summarized in Table 4**Table** .

Table 4 - Constraints

	Constraints	
Sorting System	• Input capacity	*Maximum number of drop-off facilities*
	• Bag weight and size	*Non odd-sized bags*
	• Robot capacity	*One robot can carry one bag at a time*
	• Sorting system capacity	*Equals the number of robots in the system*
	• Speed	*Considered a constant 7 km/h*
	• Number of exits	*Maximum number chutes to makeup areas*
	• Size and shape of sorting area	*Considered square or rectangular without obstacles*

Next to these constraints, the system should also comply with a number of basic conditions for operation. With a number of robots moving across the sorting area, a risk of *collisions* exist, where robots can collide against each other or an object in the sorting area, which is undesirable as it can cause damages to robots and bags and can slow the operation down. However, when trying to create a situation in which no collisions occur, the system can hit a situation in which all individual and operational robots wait for each other to take action, resulting in a *deadlock situation*. In this situation, the system reaches a state in which all robots are waiting for each other and where no robot is able to change its waiting and indecisive state. In such a situation, external intervention is needed to resolve this situation. A deadlock situation is considered undesirable, especially when it lasts long, leading to a situation where bags are not delivered to their correct makeup station in time, resulting in bags being mishandled as they can't be loaded into the aircraft in time. Next to these two basic conditions, a third seemingly simpler basic condition can be identified. In a sorting process with individual and autonomous transport robots, robots need to make sure they have enough battery power to complete a trip. A trip involves moving from the charging and storage area to the entrance of the sorting area where they can pick up bags, transporting these bags to the correct chute to a makeup station and returning to a charging and storage area. Included in this trip can be waiting time if a robot needs to wait for a bag to arrive, or when it has to wait to drop the bag at a chute when another robot needs to perform this task first. The battery of the robot needs to be sufficiently charged at the start of a trip to complete it. The final basic condition therefore formulates that battery power outs of a robot are not allowed when it is operational, meaning the robot is not allowed to come to a standstill during a trip due to insufficient battery power. The three basic conditions are summarized in Table 5.

Table 5 - Basic Conditions

	Basic Conditions
Sorting System	• No collisions between robots and robots and objects can occur • No long-term deadlock situations can occur • No power outs of robots are allowed during trips

3.4.4. Key Performance Indicators and Related Stakeholders

Key performance indicators make measurable if the use of individual and autonomous transport robots complies with the requirements described in section 3.1 and section 3 and constraints described and 3.4.3. KPIs are relevant for evaluating the baggage handling system where autonomous and individual transport robots are used for the sorting process. The values of the KPIs must show a value equal to or better than values for these KPIs in the sorting process of a conventional baggage handling system. When this new system performs as well or even better than the conventional version, the feasibility is proved, as the performance equals or exceeds the conventional version, while the new system provides more flexibility in capacity by using individual and autonomous transport robots. Key performance indicators that are relevant in measuring the performance of the new sorting system for baggage handling are indicated in Table 6.

Table 6 - Key Performance Indicators

	Key Performance Indicators	Unit	Optimal value
Sorting System	• Average process time of bags	[time]	6 minutes
	• Percentage of bags exceeding norm time	[%]	0%-0.02%
	• Average number of robots	[#]	As low as possible
	• Occupancy rate		
	– Percentage of operational time while being loaded	[%]	As high as possible
	– Percentage of operational but empty trips	[%]	A value as close to 50% as possible
	• Percentage of charging time	[%]	As low as possible
	• Number of conflicts avoided	[#]	As low as possible

The most important key performance indicators are the average process time and the percentage of bags exceeding norm time. The *average process time of bags* means the time it takes a robot to transport a bag from the entrance to the exit of the sorting area. This also includes possible waiting time of a robot transporting a bag. Based on existing practice at a medium-sized regional airport in the Netherlands, the value of this KPI should not exceed six minutes. As the floor size of the sorting system can be adjusted, some slack is allowed for this value. The optimal value for this KPI depends on the size of the floor plan but to keep the number of mishandled bags as low as in practice; a maximum value of 10 minutes is considered acceptable for this KPI. Furthermore, the deviation around the mean value of this KPI can give valuable insights into the precision and the accuracy in generating process times, measuring this part of the performance of the baggage robot concept.

This average process time is related to the second key performance indicator, the *number of bags exceeding norm time*. Exceeding the norm time means that a bag arrives at a correct makeup station too late, and therefore cannot be loaded on the aircraft before departure, resulting in bags being marked as mishandled. The percentage of mishandled bags should be as low as possible for a baggage handling system to be accepted for operation by ground handling parties and airports – depending on the organizational structure of the airport. The unit of this KPI is percentage. Following from a case study at a medium-sized regional airport in the Netherlands, 2 out of 10.000 bags being mishandled is considered acceptable, corresponding to 0.02%. The optimal value for this KPI is 0% but values between 0% and 0.02% are considered acceptable.

Intermezzo: Definition mishandled bags

A mishandled bag is defined as checked baggage that is delayed, damaged, pilfered, lost or stolen (SITA, n.d.). This research does not consider damaged, pilfered, lost or stolen bags. This research uses the term 'mishandled' to mark bags that are handled by the baggage handling system but arrive at a makeup station too late, which means the bag cannot be loaded into the aircraft. This results in a situation in which the bag needs to be sent to the associated passenger at a different time or a different way, resulting in a delay. This research assumes bag can't get lost within the baggage handling system.

When these two KPIs have values that are equal or better than in the sorting process of existing baggage handling systems the new concept is considered feasible. Next to these main KPIs, four additional key performance indicators can be considered to get a better picture of the performance of the new concept.

The first is the *average number of robots*. The number of robots should be sparse. It should be no more and no less than the number of robots required for handling checked baggage at peak demand. The unit of this KPI is an integer number [0, .., n]. The optimal value of this KPI is the least number of robots under the condition that the system complies with all requirements and constraints.

Another KPI to be considered is the occupancy rate, which can be split in two sub-KPIs, both expressed in percentages. The first sub-KPI of occupancy rate is the *percentage of time robots are operational and loaded*. Robots can be operational and non-operational. Non-operational time includes storing and charging time. This KPI shows the percentage of the total time that a robot is executing a transport task. A low value of this KPI can indicate several things, including having too many robots in the system and inefficient charging strategies. The second sub-KPI of occupancy rate is the *percentage of operational but empty trips*. When a robot is operational, it can be either loaded or empty. The percentage of empty trips should be minimal, indicating an efficient allocation of robots to bags. An ideal value is 0% but unrealistic, as bags are only transported from entrance to exit and not the other way around. A value of 50% therefore shows the maximum utilization of robots, representing one loaded trip from entrance to exit and one empty trip from the exit back to the entrance to pick up a new bag. A value of 50% however is also unrealistic, as robots also need to make trips to the charging and storing area when they run out of battery power or are idle for some time during for example the night. A value that comes close to this 50% is therefore considered optimal.

The third additional KPI is the *percentage of charging time*. This KPI refers to charging efficiency. The amount of time that a robot spends charging should be as low as possible, still ensuring effective operation. The unit of this KPI is percentage and the value should be as low as possible, under the condition that the system complies with all the requirements and constraints of the system.

The final additional KPI is the *number of conflicts avoided*. As described earlier, conflicts and deadlocks might occur in the system due to decentralized control. However, conflicts can be avoided, before escalating to an actual conflict which may lead to system deadlock. The number of conflicts that are avoided is a KPI to measure the performance of the new sorting concept and should be as low as possible. A low value of this KPI indicates that the configuration of the system causes not too many imminent conflicts. However, a low value is not as strict requirement, as long as all the potential conflicts are being avoided and does not cause too much delay in the transportation process, causing an increase in the percentage of mishandled bags.

All mentioned key performance indicators can be linked to relevant stakeholders in the baggage robot concept. The four most relevant stakeholders are airlines, airports, ground handlers and passengers. Appendix B shows the results of a stakeholder analysis. In this analysis, an overview of the problem formulations on baggage handling systems in general of the different stakeholders is provided. By making the interests and objectives of the stakeholders explicit and comparing the objectives to the current situation as perceived by the different stakeholders, these problem formulations can show a gap.

Appendix B shows several terms used in the stakeholder analysis: interests, desired situation/objectives, existing or expected situation and gap, causes and possible solutions. The term interest refers to the issues that matter most to the stakeholder when it comes to baggage handling systems at airports. Identifying the interest of the different stakeholders helps in estimating to what extant certain objectives or solutions will be acceptable for that specific stakeholder. The objectives or desired situation describe what the stakeholders wish to achieve when it comes to baggage handling systems. Objectives can be used as a measure to judge the existing situation, as it can indicate a gap between the objectives or desired situation and the perceived existing or expected situation. Often, this gap can help in making the nature of the problem more explicit (Enserink et al., 2010).

As becomes apparent from the stakeholder analysis, all relevant stakeholders wish to have a baggage handling system that has an outstanding performance, but the motivation for this wish differs. For example airlines wish that baggage handling systems are as flawless as possible to reduce their expenses on compensation for lost or delayed bags, while airports don't want a poorly performing baggage handling system to harm their reputation. Ground handlers wish to have an outstanding baggage handling system to have an advantageous competitive position in tendering processes, while passengers wish their bags are at the right reclaim belt at the right destination at the right time to leave the airport carefree. For the airlines, airports and passengers, having a baggage handling system that reduces the chances of bags getting mishandled or delayed help in fulfilling their wishes. For ground handlers however, this solution is only partially fulfilling their wish, as they can obtain an advantage over their competitors in tendering processes if they can dynamically alter the capacity of the baggage handling system to minimize the operational costs of the system, while guaranteeing accurate performance.

When it comes to the described KPIs, the owner of the baggage handling system is interested in all of the seven KPIs. Depending on the formal structure of the airport – which can be different for each airport – the baggage handling facilities are owned by a party. The two parties that are found to own the infrastructure most often are the airport and the ground handling parties. For the other involved stakeholders – passengers and airlines – the most important KPI is the percentage of bags exceeding norm time (or the percentage of mishandled bags) and indirectly the process time of bags (related to the percentage of mishandled bags) and the average number of robots (related to the operational costs for the owner of the system, who passes the costs on, eventually resulting in an increase in ticket prices). When it comes to baggage handling systems, the most important thing for passengers is that bags are loaded into the same aircraft as the accompanying passenger. For airlines this is also the case, but more indirectly as having a high percentage of bags exceeding the norm time means the airlines have to compensate the duped passengers. All KPIs can be combined with one or more stakeholders:

- Average process time of bags
    - Direct: owner of baggage handling system (airport and/or ground handler)
    - Indirect: airlines and passengers
- Percentage of bags exceeding norm time
    - Owner of baggage handling system (airport and/or ground handler), airlines and passengers
- Average number of robots
    - Direct: owner of baggage handling system (airport and/or ground handler)
    - Indirect: airlines and passengers
- Percentage of operational time while being loaded
    - Owner of baggage handling system (airport and/or ground handler)
- Percentage of operational but empty trips
    - Owner of baggage handling system (airport and/or ground handler)
- Percentage of charging time
    - Owner of baggage handling system (airport and/or ground handler)
- Number of conflicts avoided
    - Owner of baggage handling system (airport and/or ground handler)

3.5. Conclusion: Towards Testing the Baggage Robot Concept

The baggage robot concept tries to eliminate the need for conveyor belts in order to make the baggage handling system more scalable and dynamic. To research whether or not such a system could be feasible, it needs to be build and tested. This chapter identified the most important elements of the baggage robot concept, being the floor layout configuration and charging of the robots, the control of the system, the routing of the robots within the baggage handling area and collision and deadlock avoidance.

Floor layout configurations can differ by altering the locations of the charging and storage positions that are used by the robots to charge their battery by means of inductive charging. Furthermore, the hybrid control architecture is identified as the most suitable control architecture, providing a combination of decentralized and centralized control, making it suitable for the baggage robot concept as it increases the robustness, scalability, flexibility and performance of the system. The robots in the system are routed through the baggage handling area by following individual shortest paths. For collision and deadlock avoidance, two strategies are a stopping and resuming policy for imminent side collisions, and a turn, (wait) and continue policy for imminent frontal collisions. By invoking these collision avoidance measures, deadlock situations are avoided too.

The requirements, constraints and key performance indicators of the baggage robot concept described in this chapter are important to investigate the feasibility of the concept. As constructing a test setup of this system is laborious and expensive, a simulation model of one of the processes of a baggage handling system – the sorting process – is constructed. The next chapter involves an argumentation on an appropriate simulation method as well as the simulation model building.

4. Model Building

Constructing the real-size test setup of the baggage robot concept is laborious and expensive. Using modelling and simulation a low-cost, less time-intensive evaluation of a part of the rough design can be performed. In this chapter the use of simulation in a broader sense of airport operations is discussed, after which different simulation methods follow. A choice for an appropriate simulation method is followed by a roadmap on how to construct a simulation model of the baggage robot concept, starting with the conceptualisation of the model. In the remainder of the chapter the model building steps are followed, ending with the verification and validation of the constructed simulation model.

4.1 Simulation in Airport Operations

To test and evaluate the feasibility of individual transport robots in a baggage handling environment, simulation is an appropriate method.

Smith (1998) defines simulation as:

the process of designing a model of a real or imagined system and conducting experiments with that model.

Simulation is more than making a model; it is about using the model by running simulation experiments with it. The main advantage of using simulation models is that it enables analysis of effectiveness and predictions of implications of proposed changes to an existing or a new system, without actually changing the existing system or building a new system (Wilson, 2005). Changes to an existing system or construction of a new system and performing physical experiments on it is generally very costly and time consuming. In addition, changing one element of the system while keeping other elements constant is often not possible in physical experiments (Robinson, 1969)

Simulation …

There is not just one type of simulation. According to Smith (1998) the two main types of simulation are discrete event simulation and continuous simulation. The classification is based on the way in which state variables change. When these variables change instantly at certain separate points in time, Discrete Event Simulation (DES) is mostly used to model stochastic events and variations in processes in complex systems. It is used in many different disciplines (Riley, 2013). Where DES uses distinct points in time for changing the state variables, in continuous simulation state variables change continuously over time (Smith, 1998). According to Klee (1986), continuous simulation is mainly used to simulate the behaviour of complex and dynamic systems, which DES can also do. Continuous simulation usually describes such systems using a mathematical model or function in which time can be varied. One could state that when the system has a nature of variables not changing continuously but in discrete times and by discrete steps, a discrete event simulation is best suited to model the system. When a system is characterized by continuously changing variables over time, continuous simulation is most suitable (Özgün & Barlas, 2009).

However, not only discrete event and continuous simulation are popular methods of simulating complex systems behaviour. Another widely used simulation model in different disciplines is a model that is able to describe autonomous and individual agents. This type of Agent-Based Modelling (ABM) and simulation is used for modelling complex systems, but unlike continuous and discrete event simulation, ABM uses autonomous agents that can interact with each other and have specific, modelled behaviour (Macal & North, 2010).

… in Airport Operations

Existing literature shows that for simulating airport operations in general and baggage handling system, two types of simulation are mostly used:

1. Discrete Event Simulation
2. Agent-Based Modelling

Both of them have been briefly discussed. There are multiple studies which can be categorized according to their application and type of simulation, see Table 7.

Table 7 - Overview of simulation methods and applications

Application	Discrete Event Simulation	Agent-Based Modelling
Air Cargo Operations	(Nsakanda, Turcotte, & Diaby, 2004)	(Zhu, Ludema, & van der Heijden, 2000)
Passenger Flows	(Guizzi, Murino, & Romano, 2009; Rauch & Kljajić, 2006)	(Eilon & Mathewson, 1973; Lui, Nanda, & Browne, 1972; Schultz & Fricke, 2011)
Baggage Handling Systems	(Johnstone, Le, et al., 2015; Johnstone, Creighton, & Nahavandi, 2015; Le, Zhang, Johnstone, Nahavandi, & Creighton, 2012; Savrasovs, Medvedev, & Sincova, 2009)	(Hallenborg & Demazeau, 2006, 2008)

4.1.1.1 Discrete Event Simulation

Discrete event simulation is a widely-used method to simulate different aspects of airport operations. Existing literature shows the application of DES for analysis and evaluation of air cargo operations, passenger flows and baggage handling systems.
Air cargo operations are considered to be highly complex. There are many interdependent processes involved and cargo comes in a wide variety of commodities. The complexity increases when a distinction is made between service types and preferred type of aircraft. The aim of Nsakanda et al. (2004) is to develop a (simulation) tool to evaluate and analyse air cargo operations. Their decision to apply discrete event simulation is based on a study by Delorme, Procter, Swaminathan, & Tillinghast (1992) on the use of simulation for airport cargo operations. Nsakanda et al. (2004) used Arena software to develop their model, with shipments as key entities being processed in the cargo terminal. The route that these shipments - represented by entities - follow is modelled as a random processing route.

Another application of DES in airport operations focuses on passenger flows. Passenger flows are everywhere in the terminal parts of airports. They start when passengers arrive to the check-in or bag-drop desk and ends with passengers queuing to board the plane. According to Rauch & Kljajić (2006), these passenger flows can be characterized as discrete stochastic processes. In combination with the frequent use of discrete event simulation for modelling complex systems with infrastructure constraints and limited capacity, DES is found to be a suitable method for modelling passenger flows at airports. In their research on using DES for modelling passenger flows, Guizzi et al. (2009) use Arena software, with passengers as key entities and check-in desks and security checkpoints represented by processes.

Although DES is applied in the airport operations domain on air cargo operations and passenger flows, it is mostly used on baggage handling systems. Research articles on this application differ in their focus. Johnstone, Creighton, et al. (2015) focus on the locations in baggage handling systems where sorting takes place, which can cause bottlenecks. They have used discrete event simulation to assess the impact of variations in the physical layout of merge points in the BHS on throughput performance. This research focuses on a very specific aspect of BHS, whereas the main author in a different research broadens this scope by using discrete event simulation to analyse baggage flows in the BHS of an international airport, to identify key factors that can influence the total dwell time of baggage items (Johnstone, Le, et al., 2015). However, Johnstone, Le, et al. (2015) are not the only ones who applied DES to the performance of BHS. Le et al. (2012) researched a broader aspect of BHS performance, where Johnstone, Le, et al. (2015) focussed on dwell time, Le et al. (2012) researched expected BHS-performance using discrete event simulation, applying different output measurements like throughput, travel and in- system time, and queuing delay to determine the performance of the system as a whole. Next to using DES to measure the BHS-performance in its current state, research has been performed on the sustainability of these BHSs. Savrasovs et al. (2009) used discrete event simulation to research - applying different scenarios - what should be improved or changed in the baggage handling systems at Riga Airport when passenger flows increase. In general, one could state that the most commonly used application of discrete event simulation in the field of baggage handling systems is to analyse the performance of the system and how future-proof the system is.

4.1.1.2 Agent-Based Modelling

Agent-Based Modelling (ABM) is a method suitable for modelling different aspects of airport operations and modelling passenger flows in particular, since ABM is able to describe individual agents that can interact with each other and can be assigned specific behaviour. However, existing literature shows that ABM is also used on baggage handling systems and - to a lesser extent - air cargo operations. This section elaborates on the use of ABM on airport operations processes.

Zhu et al. (2000) define a multi-agent system for modelling air cargo transport. In this case the agents do not represent individual human beings, but one agent represents one origin-destination flight leg with specific attributes such as capacity and space available on that leg. These agents could then communicate and cooperate with each other to assign the cargo on planned shipments.

Another application of ABM is in the field of passenger flows. Eilon & Mathewson (1973) use ABM to evaluate the design of an airport terminal building where passengers move through. Passengers moving through an area like a terminal have to be processed, which takes time, and can therefore cause congestion. These authors model passenger flows by including amongst others flight schedules, service rates and resources and individual passenger characteristics (Wu & Mengersen, 2013). A year earlier, Lui et al. (1972) used the same modelling technique to analyse passenger and baggage processes at JFK Airport, New York. In their model, agents did not represent individual passengers, but a group of passengers coming from the same flight. These groups are characterized on eleven parameters like airline carrier, flight number, group size and number of bags for the entire group (Lui et al., 1972). A more recent study by Schultz & Fricke (2011) also focused on the outbound passenger process. Contrary to Lui et al. (1972), their agents represent individual passengers which are modelled to the detail of motion speed and dynamic group behaviour based on walking speed of other agents and density.

Baggage handling systems can also be modelled and simulated with agent-based modelling. Hallenborg & Demazeau (2006) explain that baggage items are passive and dependent on the elements of the BHS to move. The authors model eight BHS-elements as agents amongst which a top loader agent bringing baggage items into the BHS, a divert agent that represents divert elements in the BHS that allows multiple routes to be taken and a merge agent merging baggage items from two input legs to one output leg. Between these agents, Hallenborg & Demazeau (2006) modelled an interaction mechanism to enable them to communicate and perform the baggage handling process. In a follow-up study of the same agent design is used, now with individual control logic to constantly adapt to the current situation of the neighbouring agent. The authors applied their multi-agent system to a real baggage handling system at a big hub in Asia (Hallenborg & Demazeau, 2008). Another application of ABM on BHS on a real airport was conducted by Cavada et al. (2017). They argued that an analysis of different operating strategies on baggage handling systems must include the elements and agents that are part of the BHS, not as isolated elements or agents, but as an integrated whole. In their use of the case of Santiago International Airport in Chile, they use microscopic vehicle traffic simulation as a basis in which the streets equal the conveyors and the vehicles equal baggage items. This unconventional use of microscopic vehicle traffic simulation appeared to be suitable for determining the capacity of this airport's BHS.

4.1.1.3 Strengths of Discrete Event Simulation and Agent-Based Modelling in Airport Operations

Discrete event simulation is a commonly used method to simulate different kinds of airport operation processes because it has specific strengths that are useful for this application. In DES, physical and complex systems are reduced to a series of processes, with entities flowing from one to another process (Johnstone, Le, et al., 2015). During this flow through different processes, as described earlier, the state of these entities only changes at discrete points in time when events emerge. For the application of discrete event simulation in airport operations, this discrete character is very useful as it resembles the true nature of passenger or bag flows processes, which are discrete and stochastic processes (Rauch & Kljajić, 2006). Combined with the frequent use of DES for modelling complex systems with limited infrastructural capacity, processes like baggage handling processes in airport operations are very well suited for discrete event simulation (Rauch & Kljajić, 2006).

The key strength of Agent-Based Modelling is that its focus is not just on individual components of a system but rather on the interactions amongst different components. By also modelling these interactions, they can be optimized as well. This differs from discrete event simulation where it is only possible to optimize one single component or the system as a whole, without taking interactions into account. In that sense, ABM is more accurate to model systems where interactions play an important role in the performance of the system, like passenger flows where passengers interact with each other and adjust their actions as a result of these interactions (Cavada et al., 2017). Another strength of ABM for airport operations is mentioned by Hallenborg & Demazeau (2006). Their proposal for future work suggests that by using agent-based modelling, less system specific calibration is required, but would like to see that proven more.

4.1.1.4 Weaknesses of Discrete Event Simulation and Agent-Based Modelling in Airport Operations

Discrete event simulation in airport related processes also has certain downsides. According to Riley (2013), a yet unsolved challenge in optimizing DES-models is to exactly determine an optimal solution for a model. This inability of DES is even more problematic when there is a big feasible solution space. New ways to optimize large scale DES models are still lacking (Riley, 2013). So, while there is a need for optimization in DES, ways to do so are still missing and big solution spaces complicate this even more. Another complication when applying DES in airport related processes is that lots of data are required to model the system accurately. Amongst others, data on mean process times as well as distribution characteristics such as process time standard deviation are needed, taking a lot of time and resources, given the complex nature of airports.

When it comes to using ABM in airport operations, it is less often used than discrete event simulation. It seems that DES is the preferred method in this field. The suspicion is that the reason for this difference is that on average, ABM is harder to apply to airport operations since not only the system needs to be modelled, but also a deep understanding of the interactions between model components is required as decision logic also must be included. Hallenborg & Demazeau (2008) provide examples of this. This means that all interaction assumptions need to be made explicit and more than only the physical system needs to be described.

4.2 Argumentation for a Simulation Method

In section 4.1, two simulation methods are described: Discrete Event Simulation and Agent-Based Modelling. To build a simulation model of a baggage handling system that uses autonomous and individual transport robots, a decision needs to be made between these simulation methods. The strengths and weaknesses of both methods have been discussed in section 1.3.1. The method of choice in this research is Agent-Based Modelling. The most important reason for this choice is the autonomous nature of the individual transport robots considered. In literature, the single most given reason on using ABM can be paraphrased as that agent-based model are able to explicitly model the complexity that arises from individual actions and interactions that arise in the real world (Siebers, Macal, Garnett, Buxton, & Pidd, 2010). Borshchev & Filippov (2004) point out that a discussion on the exact definition of agents is still going on, but he describes some properties that an object should have to be called an agent. Amongst these properties are pro- and re-activeness, spatial awareness, ability to learn, social ability and intellect. Siebers, Macal, Garnett, Buxton, & Pidd (2010) add that these agents are in fact discrete entities 'that are designed to mimic the behaviour of their real-world counterparts.'

In literature, the single most given reason on using ABM can be paraphrased as that agent-based model are able to explicitly model the complexity that arises from individual actions and interactions that arise in the real world (Siebers, Macal, Garnett, Buxton, & Pidd, 2010). The authors describe for which problems ABM is a suitable method by listing relevant features of these problems. When:

- it is important for individual agents to include (geo-)spatial aspects to their behaviour (which enables agents to move over a landscape)
- it is important that agents are able to learn or adapt
- agents engage in strategic behaviour and anticipate on the reactions of other agents' in their decision-making process
- it is important to model agents that cooperate, collude, or form organisations
- the past is not a predictor for the future (for example in the case of new technologies, like autonomous and individual transport robots)

The first feature on (geo-)spatial aspects is underlined by Dubiel & Tsimhoni (2005) who explain that ABM is more suitable for simulating the free movement of entities, opposed to DES, as the primary limitation of DES is that movements are generalized in the simulation. According to Dubiel & Tsimhoni (2005) commercial DES packages require a path between two points in order for an entity to move between these points. When one wishes to give agents the freedom to move around freely, ABM is more appropriate since every path an agent could take between two points should be modelled in order to represent a free moving space. The decisions that are made at all these points do not represent autonomous agents since all of the decisions on processes or routing must be made by central servers, which could affect the interaction between the agents of interest and the other agents and objects that are present in the simulation model (Dubiel & Tsimhoni, 2005).

In this research, the focus is on the autonomous and individual transport robots, which drive the system as a whole. These robots can be represented as agents having their own set of goals and behaviours and their own thread of control. By representing these robots as agents in an agent-based model, the modelled robots modelled are having the capability of making autonomous decisions – the robots are able to take flexible action in reaction to their environment – and the ability of showing proactive behaviour. By controlling the system decentralized, each agent has its own set of goals and behaviours and its own thread of control (Siebers, Macal, Garnett, Buxton, & Pidd, 2010).

As Janssen (2005) describes, agent-based modelling is 'the computational study of social agents as evolving systems of autonomous interacting agents'. This makes ABM suitable for researching the system of autonomous and individual transport robots from a perspective that allows for complex and adaptive systems. ABM allows for experimenting with the simulation model to explore the effects of different agent attributes, behavioural rules and types of interactions on the system on a macro level. In their work, Borshchev & Filippov (2004) describe that agents can model 'objects of very diverse nature and scale'. On the more physical level, pedestrians, cars or robot can be modelled as agents in Agent-Based Modelling.

The control of an agent-based model is decentralized. Borshchev & Filippov (2004) describe this decentralized type of control as an important feature of ABM. The behaviour can be defined at the level of individual agents and the global behaviour, which can be an optimization, emerges as a result of the behaviour of all the agents present in the system, each following their own decision rules while continuously communicating with each other and the environment.

Opposed to Discrete Event Simulation, Agent-Based Modelling enables the researcher to model and explore the interactions between both agents and agents, as agents with the environment. In the case of this research, transport robot to transport robot interactions could be modelled and explored, as well as the way in which the transport robots interact with their environment, like walls, bags and areas. In their interaction with the environment, agents are capable of deriving information from this environment, which they use to make form their perception about the state of the environment at that moment. Janssen (2005) uses robots as an example in his argument that agent-based modelling enables a researcher to research and explore the behaviour of adaptive autonomous agents in the physical world. According to Siebers, Macal, Garnett, Buxton, & Pidd (2010), agent-based modelling is being applied to the more traditional operations front like dynamic supply chains, but only when these operations require that the modelling includes dynamic processes that are able to quickly adapt to changing requirements and events on a real-time basis. Discrete Event Simulation is not easily able of processing decisions made at very small time increments. It can be done, by placing a large amount of decision points extremely close to one another and including decision logic in each of these points. Next to this being a very laborious process, the results of would be hard to verify and validate (Dubiel & Tsimhoni, 2005).

The flexible and autonomous nature of autonomous and individual transport robots can be modelled with this method. Wooldridge (2002) explains flexibility of agents as the goal-directed nature of agents, combined with their ability to be reactive and their capability of interacting with other agents in the system. The goal-directed nature can for example be that agents work to maximize their utility and reactivity means as much as the ability of the agents to respond to changes in the environment. Agents use the combination of their goal-directed nature and attributes for their decision-making process, in which decisions on their actions are made as well as how these actions influence the environment (Janssen, 2005). Another useful aspect of ABM is that agents are able to interact indirectly. This type of interaction can be used to exchange information about possible strategies and knowledge about resources and agreements in an effort to solve collective problems (Janssen, 2005). ABM attempts to simulate intelligent and autonomous agents as they interact to accomplish a goal in their environment (Dubiel & Tsimhoni, 2005).

The limitations of DES and the advantages of ABM combined result in ABM being the preferred simulation method. The limitations of DES can be overcome, but it does not make it the most suited method of representing autonomous and individual transport robots that have certain decision patterns.

4.3 Modelling Steps

The goal of the simulation model is to provide the user of the model with insights into the performance of a baggage handling system that uses autonomous and individual transport robots for the sorting job of the system. The modelling steps to develop a correct simulation model are shown in Figure 19. The steps are derived from Marion, Scotland, Lawson, & Marion (2008) and Maki & Thompson (2005).

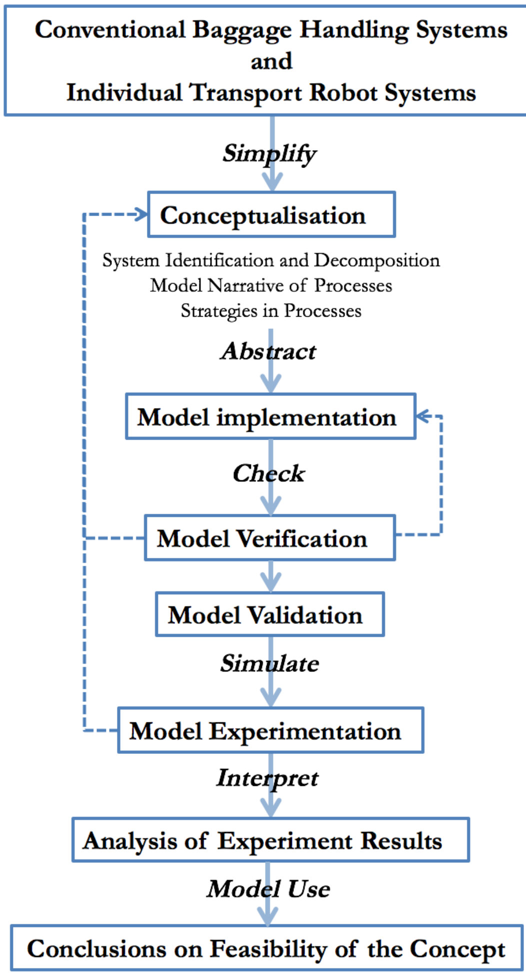

Figure 19 - Modelling Steps

Developing a simulation model starts by having a deep understanding of the system to be modelled. Since the system to be modelled does not exist in the real world yet but rather an advanced combination of two systems – a conventional baggage handling system and an autonomous and individual transport robot system – both systems need to be understood. As there is no fully autonomous and individual transport robot system in operation to the knowledge of the researcher, instead of understanding the autonomous system, an understanding of the differences between an AGV system and an autonomous system is necessary. The phase of understanding these systems is performed in Chapter 2. The next step is to simplify the combination of these complex real-world systems by means of conceptualisation. This conceptualisation entails the identification and decomposition of the system to be modelled and clear insights into the processes that are relevant in a baggage handling system where the sorting process is executed by autonomous and individual transport robots. In the sorting process, several strategies play a role, such as choosing a route, choosing dispatching rules, charging strategies and strategies to avoid collisions. After all elements that are relevant for accurately modelling the system are identified, the conceptualisation can be abstracted in the model implementation phase. After implementing the system in software, the simulation model needs to be checked by means of verification – is the system built right? – and validation – is the right system built? After the simulation model is checked, experiments can be executed with it. Finally, the results of the experiments executed can be analysed to make statements on the feasibility of the autonomous and individual transport robot system in a baggage handling environment.

4.4 Model Conceptualisation

The first modelling step concerns the model conceptualisation. This section starts with the scope of the model to be built, after which requirements, constraints and assumptions for the simulation model are described. After that, the baggage robot system elements are identified and the sorting process of the system is decomposed. The section ends with a description on how the most important processes in the baggage robot concept as described in section 3.3 will be translated into a the simulation model.

4.4.1 Model Scope

In the most aggregated view on baggage handling systems, three main processes that bags have to go through in their journey from bag drop-off to loading into the aircraft can be identified: (1) security screening, (2) sorting and (3) makeup (and storing) of baggage items. In order for a bag to arrive at the security screening, it first needs to enter the baggage handling area. After the makeup and storing process, the bags exit the baggage handling area to be loaded into an aircraft. Since this research is limited to the size of a regional airport operating in a point-to-point network, transfers are not included. If transfers were to be included, transfer bags would be reinserted into the system at the security screening process.

In between the key processes of handling baggage, baggage items need to be transported between these processes. The four key transport moments are depicted in Figure 20. The conventional baggage handling system uses conveyor belts to take care of transporting bags between processes.

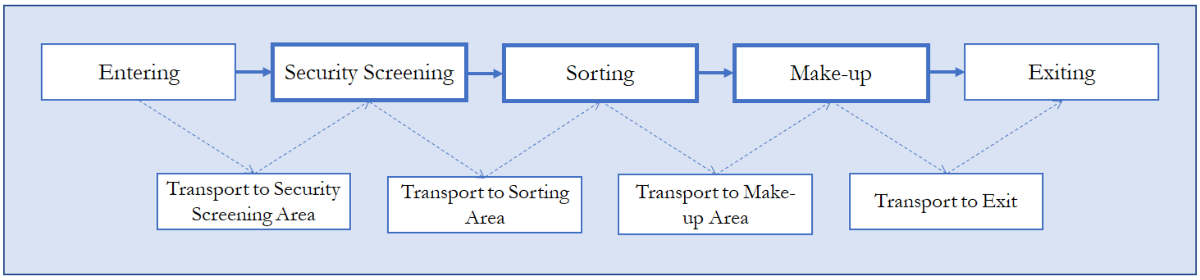

Figure 20 - Key processes in baggage handling at regional airports operating in a point-to-point network

This research tries to identify how autonomous and individual transport robots can help in dynamically altering the floor plan and desired capacity of baggage handling system. To do so, a simulation model is constructed. Simulation models are found valuable in explaining and understanding real-world phenomena. Phenomena that are costly or impossible to perform in a laboratory, or difficult to collect in field experiments, are especially suited for simulation studies, as is the case for the baggage robot concept. However, from a performance concern point of view it is impossible and unnecessary to include all elements, amongst which are elements that do not have much effect on the system, into the simulation model (Xiang, Kennedy, & Madey, 2005). The simulation model should at least be able to represent the physical baggage robot concept system, to enhance the understanding of the system and to predict and control the behaviour of the baggage robots in the system under different circumstances. This is why a clear delineation is necessary. As the individual transport robot system can be represented by an agent system, key processes from Figure 20 that can be represented by agents are identified.

The processes 'entering' and 'security screening' can be categorized as being queuing systems which can be represented by discrete event models. Since the interest of this research is to investigate the potential of agent systems in a baggage handling context, these elements are eliminated from the scope of the simulation model to be build. The hypothesis of this research is that the biggest lead time improvements by making use of autonomous and individual transport robots can be achieved in the sorting system, which can be categorized as an agent system. The processes 'storing' and 'exiting' are also eliminated from the scope as they are performed by either humans or robot arms, which are other types of systems. The key processes that are included in the model building are indicated in Figure 21.

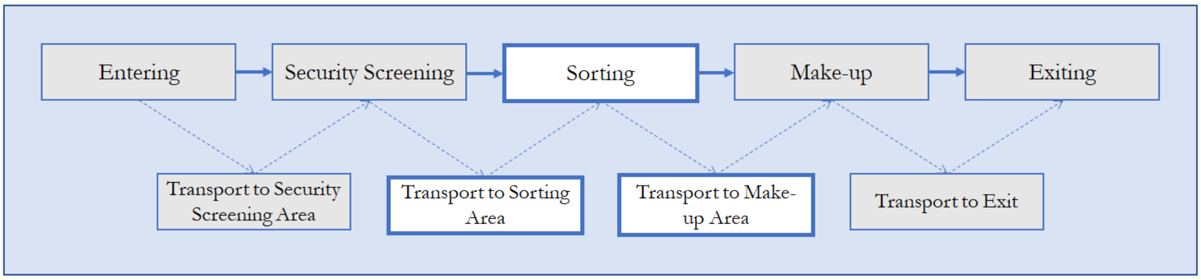

Figure 21 - Scope of Model

Now the scope of the model is identified, it's time to establish to which requirements, constraints and assumptions the model needs to comply.

4.4.2 Model Requirements, Constraints and Assumptions

As the simulation model does not include all the possible details and behaviours of a real baggage robot concept that is not available yet, some assumptions and abstractions must be made about the system for the construction of the model. These assumptions and abstractions however, introduce inaccuracies to the simulation model (Xiang et al., 2005). It is good to be aware of the presence of inaccuracy as constructing requirements, constraints and assumptions are inevitable in the process of building a simulation model to show the idea of the baggage robot concept.

Requirements, constraints and assumptions reduce the complexity of the model. Rather than modelling the entire baggage robot concept and its functionalities, the simulation model focuses on the development of a proof of concept of the baggage robot concept that is worth further research. In the model, the attention is on the path planning of the robots, collision avoidance measures and the effect of different layouts on the KPIs of the baggage robot concept. As mentioned in the scope, only the sorting part of the baggage handling process is concerned. The requirements, constraints and assumptions pointed out below are set around this scope.

Requirements

- Minimize bag waiting time by locating CS areas as close to the incoming conveyor belts
- Maximize the reachability of CS area positions by giving each position at least one free neighbour to move over

Constraint	**Unit**	**Value**
• Number of incoming conveyor belts	[#]	4
• Number of makeup stations	[#]	4
• Number of charging and storage positions	[#]	96
• Number of robots	[#]	1-96
• Total floor area	[m^2]	6080
• Floor area charging and storage positions	[m^2]	384
• Floor area occupied by incoming conveyor belts	[m^2]	32
• Unoccupied floor area	[m^2]	5664

• Bag dimensions	[cm x cm]	100 x 75
• Robot dimensions	[cm x cm]	150 x 150
• Robot speed	[m/s]	2
• Robot capacity	[# bags]	1

Assumptions

- Bags are cleared once they enter the modelled area
- Arrival pattern according to a normal distribution with a mean of 40 minutes and a standard deviation of 30 minutes.
- The buffer capacity at the conveyor belts that brings the bags to the sorting process area is infinite
- A makeup station can only by used by one destination at a time
- Robots are homogenous and therefore have identical capabilities

Keeping these requirements, constraints and assumptions in mind, an identification of the system to be modelled and the decomposition of this system follow in the next section.

4.4.3 System Identification and Decomposition

By following the structuring method of Dam et al. (2010), the system identification and decomposition can be established. This includes an insight into the system elements agents and objects. For agents, the actions they can perform that affect their own state or that the agents take for communicating with other agents or objects are specified. Finally, a description of the environment of the system elements is presented.

After formulating model requirements, constraints and assumptions, the next step towards building a simulation model baggage robot concept is deciding on the system composition and boundaries. By identifying the internal structure, the baggage robot concept is considered as a collection of agents and all of their interactions over time. What specifically should be identified are actors and objects that are represented in the model and the states or properties that these actors and objects can have, as well as interactions and flows present in the system. In this decomposition phase, agents, objects, their properties and interactions possibly occurring baggage robot concept are included. Dam et al. (2010) describe a structured way to identify and decompose the system under study.

In the identification and decomposition process, a distinction is made between agents and objects. Agents are recognized by their boundaries, states, behaviours and ability to interact. They are capable of making decisions independently; all other entities present are considered to be objects. For the baggage robot concept the following agents and objects exist:

System element	
• Robots	Agents
• Bags	Objects
• Incoming conveyor belts	Objects
• Chutes to makeup stations	Objects

As mentioned in section 4.2., the exact definition of an agent is still under discussion, but properties that are attributed to agents are amongst others pro- and re-activeness, spatial awareness and social ability. Robots are therefore the only true agents in this system as they are able to act autonomously and react to the actions of other robots in the system. Furthermore they are capable of planning and re-planning their own paths while communicating with other robots. Bags do not have any of these properties and are therefore classified as objects. However, they are not fixed objects as they move over the conveyor belts to enter the system and are transported by robots to the makeup stations. The incoming conveyor belts and the chutes to the makeup stations are also system elements classified as objects – as they are incapable of making decisions independently – but opposed to bags that are moved through the system, these belts and chutes are fixed.

Agents can be described and specified by properties that are regarded as states. Interactions take place between agents or between agents and objects. This differs from behaviours, as behaviours identify state changes that are caused by or lead to interactions or other state changes (Dam et al., 2010). The states, interactions and behaviours elaborated on in this section are simplified as much as possible, while attempting to not abolish relevant aspects.

The only agents in the baggage robot concept are robots. The properties that describe and specify these robots – their states – are simplified to basic internal states. The states a robot can be in are the following:

- Available (battery-level, bag assigned 0, loaded bag 0)
- Incoming request
- Arrived at belt
- Transporting bag
- Delivered bag
- En route to a storage and charging position
- Charging

States changes by tasks as:

- Pick up an incoming bag at one of the incoming conveyor belts
- Check destination of bag
- Drop an incoming bag at one of the chutes to a makeup station
- Check battery level

Robots have to transport incoming bags from one of the incoming conveyor belts to one of the chutes to the makeup stations. Bags arrive at one of the four incoming conveyor belts. These conveyor belts transports the bags to the sorting area, where the bags need to wait for a robot to pick them up. As bags are static objects, they can't independently go through the sorting area and are dependent on a robot to pick them up. Despite being a static object, bags can have states as well. However, they are not able to change these states by performing tasks themselves, as robots can. States that a bag can be in are the following:

- Approaching the conveyor belt
- On a conveyor belt
- At the end of a conveyor belt and no robot is assigned yet
- At the end of a conveyor belt and waiting for pick up by the assigned robot
- On a robot
- Delivered to a chute to a makeup station

The states of a bag can only change by the interaction with other system elements. When bags arrive at one of the four incoming conveyor belts they have the state 'approaching the conveyor belt'. When it reaches a conveyor belt, it interacts with that conveyor belt and the bag's status changes to 'on a conveyor belt'. As it moves over this conveyor belt, its state can change twice to 'at the end of a conveyor belt and no robot is assigned yet', meaning the bag is waiting until a robot is assigned to the transportation task of transporting that bag to one of the makeup stations, or to 'at the end of a conveyor belt and waiting for pick up by the assigned robot' when it waits to be picked up by the robot it's assigned to. This last state change happens through communication between the bag and a robot. When a robot picks up the bag, it reads the destination information on the bag tag of the bag and translates this into one of the four available makeup stations. The exchange of information between the bag and the robot causes the bag to change its status to 'on a robot' and the robot now knows which makeup station to head to. Once the robot drops the bag at one of the makeup stations, communication between the bag and the makeup station take place which causes the bag to change its state to 'delivered to a chute to a makeup process'.

The robots continuously interact with each other, bags, the incoming conveyor belts and the chutes to the makeup stations. This communication or interaction takes place in an environment. This environment contains all the information that is external to agents, but agents can use this information when making decisions. The environment contains everything that affects an agent, but is not the agent itself (Dam et al., 2010). This means that the environment provides exogenous variables that affect the system elements, but the environment cannot be affected by these system elements in turn.

In the simulation model of the baggage robot concept, the environment provides the following:

- The locations of the incoming conveyor belts and makeup stations
- The location of other static pre-defined features such as the outer walls of the area
- The random distribution of incoming bags across the four incoming conveyor belts

The environment interacts with the system elements, the agents and objects. However, agents and objects are not able to interact with the environment directly. Agents – robots here – interact with agents and objects alone, meaning robots interact with other robots, bags, the incoming conveyor belts and the chutes to the makeup stations. Robots communicate with other robots to avoid collisions between each other, they communicate with bags to receive information on their destination and with the incoming conveyor belts to receive a transportation task. Finally robots communicate with the chutes to the makeup stations to check whether the destination on the bag tag of the bag corresponds with the destination of the makeup stations, to ensure the correct sorting of bags.

4.4.4. Processes in the Baggage Robot Concept Model

Now the scope of the model, the model requirements, constraints, assumptions and the system elements are identified, it's time to establish who does what and when in the system to show the role of individual transport robots in the model. The behaviour of agents in an agent-based model can be captured in a story, making clear which agent does what, with whom and when. A narrative of how agents in the model act and interact helps in giving insights in how the system works. (Dam et al., 2010). The narrative represents a rough outline rather than a detailed description of how the agents and objects interact, so only the actions per time tick are roughly included in the narrative:

A cleared bag arrives at the entrance of the sorting system where it is picked up by an individual transport robot. Once picked up, the destination information located on the baggage label is read by the transport robot. With this information, the transport robot proceeds to one of the makeup and storage areas present. Each makeup station and storage area represents a destination and the transport robots match this destination with the destination of the bag they transport. If there is a positive match between a particular makeup station and storage area and a bag, the robot brings the bag to this makeup and storage area where it will drop the bag. If there is no match, the robot proceeds to a general storage area where all bags that do not yet match with a destination area are temporarily stored until the right destination area is available.

Transport robots continuously monitor their battery power. Once their battery reaches a certain threshold for minimum required battery, it is no longer allowed to start picking up bags and it needs to move to a charging station to recharge. Instead of moving to a location where it is needed most when it has no load in the process, it can also proceed to a recharge station. When the transport robot is in an idle state for longer than a certain threshold, it can move itself to a robot storage area to wait for work or to a charging station if necessary.

What these roughly described processes do has been described earlier in section 3.3. The representation of processes in the model is discussed point by point in the next sections.

4.4.4.1. Control

As described in section 3.3.2. the preferred control approach for the baggage robot concept as a whole is hybrid control. In the simulation model that focuses on the sorting process of the baggage robot concept incorporates this type of control. The hybrid control as modelled is comparable to a taxi centre. When a person calls the taxi company, it tells the taxi company where he or she is and where they want to go. The taxi centre then checks which taxi driver is closest to the person requesting a taxi ride and is not occupied yet and sends that closest available taxi to the person calling. From that moment, the taxi driver takes over the transportation request and plans its route to that person and the route from that person to its desired destination. Once the taxi driver dropped the person off, it marks itself as available again so the taxi central knows that taxi is available again for a new request. This process is visualized in Figure 22. In this image the part where centralized control is used is circled with blue and the decentralized control is marked with a green circle. The same logic is applied to the baggage robot concept. When sensors detect a bag at the end of an incoming conveyor belt, it sends out a signal to a central control unit. This central control unit assigns the closest available and sufficiently charged robot to the transportation task of transporting the detected bag to a makeup station (centralized control). Once the robot received the task, it independently performs the transportation task, while continuously being aware of its environment (decentralized control). While executing the transportation tasks, robots have to obey to the 'traffic rules' that are centrally defined. This is an example of how local communication is combined with central rules. Once the robot delivered the bag it drives back to a free storage and charging position and when it is charged enough it sends out a signal to the central control unit that it is available again and ready to receive a new transportation task. This combination of centralized and decentralized control components in the model shows how the hybrid control approach is implemented in the model.

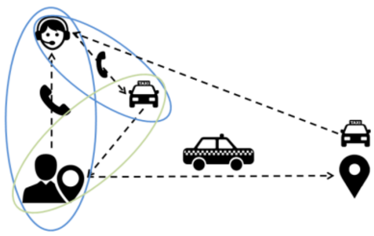

Figure 22 - Taxi Transportation Request (Image Author, 2018)

4.4.4.2. Layout

The baggage handling area in this research is considered to have fixed dimensions. The location of basic elements such as incoming conveyor belts and entrances to makeup stations are fixed as well. However, the location where individual transport robots are stored and simultaneously charged by means of drip-feeding can be altered. This research considered four layout options which are depicted in Figure 23. Green squares indicate the location of the charging and storage area. Orange squares show the charging or storage position that is either closest or furthest away from the incoming conveyor belts. Grey squares represent the shortest path possible from the closest robot charging and storage position to the incoming conveyor belts. Blue squares show the shortest path from the charging and storage position that is farthest away from the conveyor belts to these conveyor belts. The dotted line indicates the division of the layout in a north and a south part.

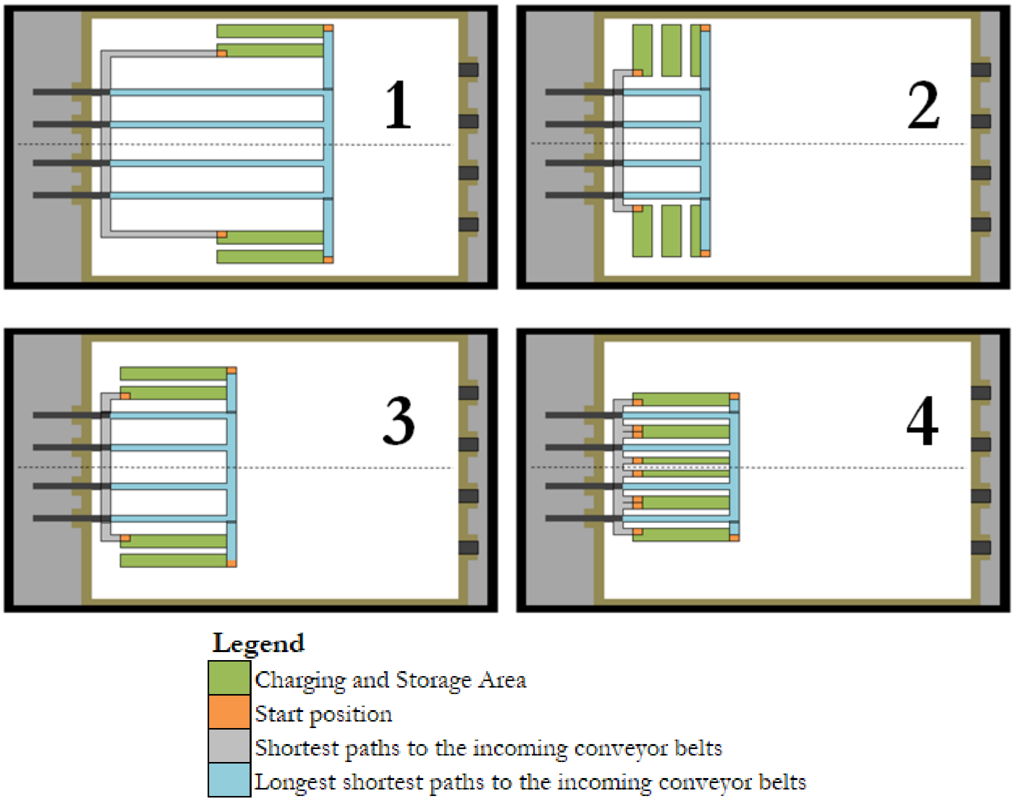

Figure 23 - Four layout configurations

These are not the only layouts imaginable but rather are designed to demonstrate how the location of storage and charging areas affect the performance of the system as a whole. Corresponding to the first design requirement – minimizing bag waiting time by locating charging and storage areas as close to the incoming conveyor belts as possible – layouts 2 to 4 are chosen. The first layout serves as a comparison, to check whether or not positioning the storage and charging areas as close to the incoming conveyor belts as possible has a significant influence on the waiting time of bags, opposed to the layout where the storage and charging positions are not located as close to the belts as possible.

The layout options are numbered one to four. These layouts are integrated in the simulation model where the numbers correspond with the following configuration names:

1. Central areas
2. Six Vertical Areas
3. Four Horizontal Areas
4. Six Horizontal Areas

For each layout the values for the shortest paths and longest shortest paths to the incoming conveyor belts are calculated. The values are visible in Appendix C.1. Belt 1 represents the nearest conveyor belt as seen from the perspective of the charging and storing area position. An example to illustrate this logic is as follows: in the first configurations where four charging areas are located in the outer north and south parts of the baggage handling area, the conveyor belt that is closest to the robots located at the northern positions is the most norther conveyor belt. This belt is therefore indicated as belt 1. The conveyor belt that is located most south is the furthest away from the northern charging and storing positions and is therefore indicated as belt 4. However, since the layouts are horizontally symmetric, the same logic can be applied to the southern charging and storing positions. Both viewpoints result in the exact same value for the shortest and longest shortest paths. As expected, the first configuration has the highest values for the shortest and longest shortest path – having minimum values of 18 and 33 respectively – as the charging and storage positions are located furthest away from the incoming conveyor belts compared to the three other configurations. The values for the shortest and longest shortest paths of configuration two and three are almost identical – 5 for the shortest path of both layouts, 19 for the longest shortest path of configuration two and 20 for the third configuration. The fourth configuration seems to be the best configuration when the aim is to minimize the travel time between charging and storage positions and incoming conveyor belts with the minimum length for the shortest path being 4 and 15 for the longest shortest path.

4.4.4.3. Charging

A robot can have several phases. It can stand idle without charging, it can stand idle and charge itself, it can drive with a bag loaded onto it and it can drive empty. Each of these statuses can have a different effect on the battery level of a robot. Robots continuously monitor their battery level. For example robots that have returned from their first transportation job go back to the storing and charging area. When it reaches this area the robot evaluates its current battery level. If this level is larger than a chosen threshold the robot will change its phase to "available". When this level is below the threshold, it changes its phase to "charging". A robot with a phase being "available" can be signalled by the system to start a transportation job. A robot that is in a charging face cannot be called while it is charging. Only when the battery level of that specific robot exceeds the set battery level threshold, it will change its phase to "available" and can be signalled for a transportation job.

The different rates in which the battery level increases or decreases differs per possible state and are showed in Table 8. This section concerns a brief explanation of these values presented.

Table 8 - Values per charging status

Status	Charging rate
Idle and not charging	- 0.0014 % /sec
Driving without a load	- 0.0042 % /sec
Driving with a load	- 0.0056 % / sec
Idle and charging	+ 0.0069 % / sec

The underlying assumptions to these values are that a robot should be fully charged within four hours and that a battery should last for at least 20 hours when the robot is unused and not charging. Furthermore it is assumed that driving without a load requires three times as much battery power and driving with a load consumes four times as much battery power compared to standing idle.

A threshold is set on the battery percentage for a robot to be allowed to start a new assignment. When the battery percentage is lower than the battery percentage required for starting and completing the longest path, the robot is not allowed to accept a new transport request for an incoming bag. The battery percentage required to start and complete the longest path is calculated in Appendix C.2 and set to 0.73%. To allow for some slack and possible unforeseen circumstances like encountering a queue on the longest path, the value is increased by ten percent, setting the threshold at a rounded level of 0.8%.

4.4.4.4. Transporting bags

This section describes the main processes of transporting bags with individual transport robots. The transport process starts by a robot *picking up* a bag from the conveyor belts that form the entrance of the baggage handling area. When a bag is picked up by a robot, the robot *transports* the bag to one of the makeup stations that correspond with the destination that the bag needs to go to. When it reaches the entrance of a makeup station, the bag is *unloaded* from the robot and placed on a chute that brings the bag to the makeup station.

Picking up bags

When bags are created, it is assigned a destination randomly which can be either A, B, C or D. As on an airport passengers can drop their bag at different drop-off points, the model randomly assigns bags to conveyor belts, representing this random behaviour of passengers. The bags are created according to an arrival pattern which is assumed to be normally distributed. Appendix F shows an arrival pattern that is used based on a busy day at a regional airport in the Netherlands. The standard normal distribution can have negative numbers, so for this research the normal distribution is cut off so the distribution is not open-ended. An open-ended distribution does not fit an arrival pattern for bag drop facilities as bags can only be dropped between two hours and 40 minutes before the departure of a flight. To be able to track the bags in the model, phases are defined for bags. When bags are created their initial phase is "approaching belt", as they are moving towards one of the arbitrarily chosen conveyor belts. When a bag reaches the end of the conveyor belt and is in the baggage handling area, it is detected by the system and the system sends out a signal to let the robots know that a bag arrived into the area and is requesting to be transported to a makeup station. This signal is received by robots that stand idle in the charging and storage areas. One of the robots that is closest to the conveyor belt where a bag is waiting and has the highest battery level claims the transport request and changes its phase from "available" to "incoming request" so it can no longer be called by another bag for a transportation job. This robot calculates the shortest path that is possible between its location and the location of the bag that is requesting transportation and moves along this path to the bag. When the robot reaches the belt, it changes its phase to "arrived at belt" and while it copies information stored in the bag tag attached to the bag, about the identity and location the bag needs to go to, it sets its loaded bag value to one. As a maximum of one bag is allowed per robot, having a value of one means that the robot is loaded and not able to take more bags along. Simultaneously with the robot copying information from the bag, the phase of the bag changes to "on robot". This change in status triggers a counter to start counting the time the bag spends on a robot. By doing so, information on the performance can be obtained, being the average time bags spend on robots for transport. Note that this is not equal to the total time a bag spends in the system, as that includes the time that the bag was on the belt and waiting for a robot to pick it up.

Transporting Bags

When a robot picks up a bag, it inherits the identity of the robot so it knows which bag number it is transporting. The bag tag that is attached to the bag during drop-off also includes information on the destination where the bag needs to go to. When the bag is dropped off in the time frame of two hours to 40 minutes before departure, one of the makeup stations is designated to this location. The robot that picks up the bag inherits the bag number but also reads the destination from the bag label. The robot is able to translate this destination to one of the make-up stations in the baggage handling area. For example, a passenger wants to check in a bag for its flight to Berlin and arrives at some time that is between two hours and 40 minutes before the flight departs. The baggage handling party responsible for assigning makeup-stations to flights assigned makeup-station B to that day's flight to Berlin. When the robot picks up the passengers bag and reads the bag tag it signals that the bag needs to go to Berlin and couples this information to the makeup-station assignment. Now the robot knows that it needs to transport the bag that is loaded onto him to makeup-station B. Having gathered this information, the robot adjusts its makeup station information from empty to B. The same logic is applied to all robots that pick up bags, making them able to translate bag destinations to the correct makeup station. As it is not possible for passengers to drop their bags more than two hours or less than 40 minutes before the flight departs, all incoming bags can be transported to a makeup station that corresponds with their destination. This means that bags don't have to be stored inside the baggage handling area waiting for their makeup station to open, as the corresponding makeup stations are already open when bags arrive.

Unloading bags

The four different makeup stations that are included in the model all have a unique location. When a robot reaches one of these locations, it unloads the bag. As the bag is unloaded, the robot sets its phase to "delivered bag" and starts driving back to the charging and storage area.

4.4.4.5. Routing

The problem considered is the coordination of the routes which robots take to transport bags in the sorting process of a baggage handling system. As multiple autonomous robots are involved, routing coordination is necessary. In other words, the problem considered is the transportation of bags between a set of fixed incoming conveyor belts O (origin) and fixed entrances to makeup stations D (destination). The transportation task or sorting of bags to a makeup station can be defined as $S^{od} \in S(t)$. For each incoming bag a robot has to be assigned to the transportation task of the incoming bag to transport the bag from its incoming conveyor belt $m \in O$ to the makeup station that corresponds with the destination the bag needs to be sorted to $n \in D$. For each individual transport robot the shortest route from the origin to the destination needs to be calculated. These shortest paths are therefore defined from the point of view of the individual robots. The objective of this problem is the following:

$$minimize \sum_{ij \epsilon A} d_{ij}$$

Subject to

$$d \geq 0$$

$$\sum_{j} d_{ij} - \sum_{j} d_{ji} = \begin{cases} 1 & if\ i = O \\ -1 & if\ i = D \\ 0 & otherwise \end{cases} \quad \forall\ i$$

In which the sorting area of the baggage handling system is considered to be an undirected graph (V, A) with source node O, destination node D and cost expressed in distance d_{ij} for each edge (i, j) in A. To find this shortest path, the A* algorithm is used. This algorithm initially was designed to improve the path planning of Shakey the Robot, which was the first mobile robot that was able to perceive and reason about its surroundings which could include obstacles (SRI International, n.d.). A* is found to be a faster version of the well-known Dijkstra algorithm as it is a best-first search algorithm. As the Dijkstra algorithm requires more computational power at every step in the simulation model, A* is found to be better for simulation purposes.

4.4.4.6. Collision and Deadlock Avoidance

Sections 3.3.4 and 3.3.5 elaborated on how the robots in the baggage robot concept make sure they don't collide into each other and how they avoid ending up in deadlocks. In the case of an imminent side collision one of the robots wait and gives way to the robot with the highest priority and when there is no difference in priority, a random one of the involved robots goes first. In case of an imminent frontal collision, the involved robots move to an empty position in the direct surrounding and recalculate their shortest path. While driving over the shortest path, that is calculated using the A* algorithm as described in section 4.4.4.5., robots continuously have to check for new possible collisions. Once a possible collision is detected, the collision avoidance measures are invoked, based on the centrally defined 'traffic rules'.

To avoid a deadlock situation, robots have the ability to communicate with other robots in their direct surrounding and one step further. When two robots are in each other's proximity and want to go to the same position at the same time, they communicate before taking a step. In this communication step, they exchange information on where they want to go and what their priority is. This thinking ahead mechanism of the robots make sure that not more than one robot can take the actual step to a position that is desired by more than one robot. This avoids collisions, but also deadlocks, as robots are able to think not only one but two steps ahead.

4.5 Model Formalisation

Now the system and corresponding agents, objects and environment have been identified, formalizing them is the next step. The concepts identified in the previous section are in a natural language and understandable for human beings, but computers have a harder time dealing with ambiguity and context dependency. This is why it is relevant to make the concepts to be modelled as specific as possible so computers are able to interpret the concepts. In other words, the system to be modelled has to be made explicit, formal and computer-understandable (Dam et al., 2010). To do so, a precise description of the concepts that play a role in the system is formulated in a non-structured list of software data structures, which converts the concepts into computer understandable analogues. Two examples of formalisation of the agents and objects present in the system are shown in Appendix D. Appendix E shows some examples of pseudo codes used to construct the model into the software in a structured way.

4.6 Software Implementation

Section 4.2 concluded that the preferred simulation method to model and simulate the baggage robot concept. The software used to implement the model is NetLogo, a java-built agent-based programming language. NetLogo provides an environment suitable for the modelling of programmable multi-agent systems. The software is open source and free to download. It also has the advantage that the software is cross-platform, in runs on Windows, Linux, Mac and other major platforms, which makes it possible to run the same model on different platforms without altering the code. NetLogo allows the modeller to provide hundreds or thousands of agents that operate independently with instructions. This feature enables the exploration of the connection between the behaviour of the individual agents on a micro-level and the overall patterns that arise from the interaction between all the agents on a macro-level (Northwestern University, n.d.). A user of a simulation model created in NetLogo can quickly and easily explore this system behaviour by altering or modifying switches, sliders, choosers and other interface elements.

In NetLogo agents (in NetLogo called 'turtles') move around in a two-dimensional world that is divided into a grid of patches. These patches have coordinates, as do turtles. However, only turtles are able to move and with that they can change their coordinates. Turtles don't move automatically, the observer needs to provide all turtles with commands and reporters so the turtles know what to do.

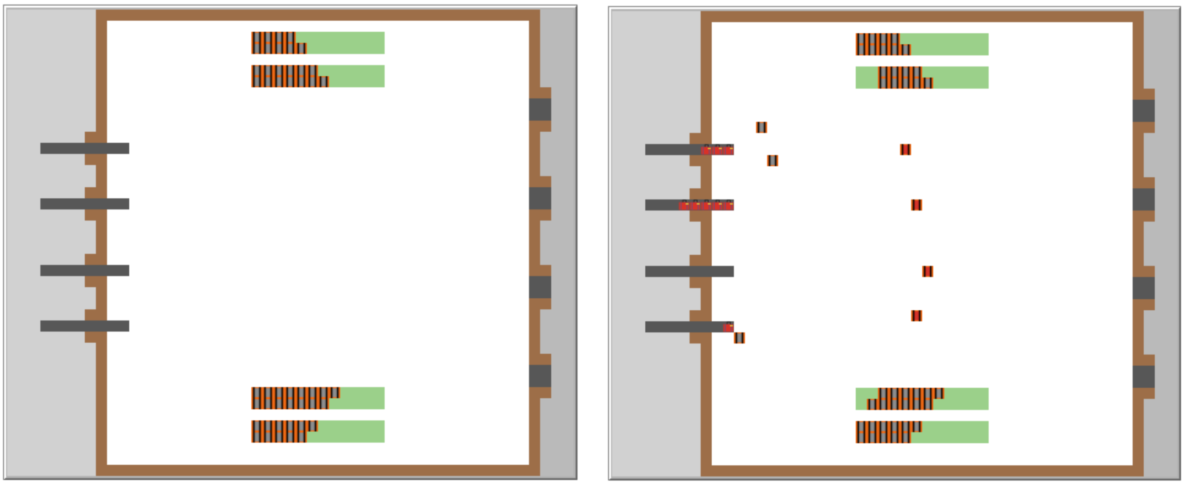

Figure 24 - The Baggage Robot Concept in NetLogo. Left: Initial Setup. Right: Agents moving in the baggage handling area

Figure 24 shows the implementation of the conceptual model in the NetLogo software. On the left the initial setup situation of the model is visible, while on the right the model is running. The orange and grey squares indicate the robots. For this figure an arbitrary number of robots – 48 – are chosen. The four dark grey horizontal lines on the left indicate the incoming conveyor belts, the four dark grey squares on the right represent four makeup stations. On the right side of the figure bags enter the system through the incoming conveyor belts.

Figure 25 - Robots and Bags

Figure 25 shows how the robots and bags are presented in the NetLogo model. On the left the initial setup of a robot is displayed. Once they pick up a bag – bags are displayed in the middle of Figure 25 – the visualisation of the robot agent changes, indicated with the right image in the figure.

The dimensions of the area displayed in Figure 24 are 20x50 patches. The white area representing the space available for the sorting process is 38x40 patches. A patch is assumed to be 2x2 meters, which means the area available for the sorting process is equivalent to 6080 m^2. The dimensions for the bags and robots are 100x75 cm and 150x150 cm respectively. Robots and bags can move through the area in discrete time steps, in NetLogo called 'ticks'. In the implementation of the sorting process of the baggage robot concept, each tick represents one second. This means that robots can move one patch in one tick, translating that to a real-time robot speed of 2 meters per second. To obtain more precision in the representation of time and space, the number of ticks that represent one second can be increased or the dimensions that a patch represent can be decreased. However, increasing the ticks/seconds ratio requires the model to run for a longer time to complete an experiment or run which can be impractical when a solid number of experiments or runs are desired.

4.7 Model Verification and Validation

The next modelling process step is the verification and validation of the developed simulation model. Verification concerns the question "is the *thing* built *right*?", whereas validation addresses the question "is the *right thing* built?". This means that verification concerns whether or not the model implementation matches the designed system and validation is used to see if the designed and constructed model helps to answer a or the research question. This section is concerned with both the verification and validation of the constructed simulation model.

4.7.1 Model Verification

Model verification is a process to check whether the model conceptualisation is implemented right in the software used (Xiang et al., 2005). A model needs to be verified before it is used. There are several model verification tests that checks if the entities and relationships as established in the model conceptualisation phase have been translated into the code of the simulation model correctly. The verification phase is especially difficult for agent-based models as a large variety and number of agents, agent states and possible interactions are possible (Dam et al., 2010).

Dam et al. (2010) describe four main parts of verifying agent-based models:

1. Recording and tracking agent behaviour
2. Single-agent testing
3. Interaction testing
4. Multi-agent testing

The first part – recording and tracking agent behaviour – is used to verify the model operation. To do so, output variables are monitored to investigate both individual agent behaviour and the operation of the model on a system level. The states of the internal processes of individual robots have been monitored and the collective robot behaviour is checked. The second part – single-agent testing – involves the exploration of single agent behaviour. By means of unit tests implemented in the model code the behaviour of robots is tested. The third part – interaction testing – tests the interaction between agents in the model. By performing theoretical predictions and checking the behaviour of the robots when they encounter each other, the basic agent interactions are checked to see if they happen as designed. The final part – multi-agent testing – verifies the entire model. In this part theoretical predictions are also used to check the behaviour of the agents within the system as a whole. These theoretical predictions can be supplemented by varying the input parameters to check if the agents act intuitive in changing circumstances as well (Dam et al., 2010).

Table 9**Table** shows an example of how each of these verification tests is executed and what their result are. For each test type a small description is given, as well the theoretical prediction of the result. After executing the test, the obtained result is reported and a conclusion is drawn whether or not the test verified that part of the model. If the obtained result differed from the expected result, corrective measures have been taken in the model code. A complete overview of all the verification tests performed can be found in Appendix I. Next to these tests, the model have been continuously checked and debugged during the model building phase. When a dysfunctional part of the code was identified, the seed of the pseudo-random number generator of NetLogo was fixed so the dysfunctional part of the code could be examined and corrective measures could be tested. An agent delete button was also implemented in the model environment to verify if the model was able to execute its desired tasks and behaviours when an unexpected situation was created with the delete button. An example of this is that bags are expected to move over the conveyor belt when the patch ahead of them was free, until they reached the end of the conveyor belt. By using the delete button the bag at the end of the conveyor belt was deleted to see if the bags behind it would notice this and move forward to occupy the unexpected free position at the end of the conveyor belt. By checking this after every improvement of the model, the bag moving behaviour was verified.

Another verification test that has been performed is the manual calculation of KPI results. For several KPIs manual calculations are performed for all four layout configurations. Appendix I shows an example of the manual calculation of the minimum value for the KPI 'average process time of bags'. The manual calculations show that for example for the KPI 'average process time of bags' for the central areas layout configuration, the minimum value of this KPI cannot be lower than 68 seconds. Running the model shows that indeed the value of 'average process time of bags' does not drop below this minimum value.

Table 9 - Examples of Model Verification Tests

Test Type	Description	Expected Result	Obtained Result	Verified?
1	Testing the phases of robots	Robots that are assigned to a transport task go through the six possible phases in order.	All robots tested successfully go through the six phases in sequence.	Yes
2	Testing the robot assignment	Only robots with the phase 'available' can be assigned to a bag that requires transportation.	All tested robots showed correct task assignment after which the phase of these robots changed from available to 'incoming request'.	Yes
3	Testing the priority rules	Robots with a value of 0 for the priority attribute have to give way to a robot with a priority value of 1 in case of an imminent collision.	The giving way behaviour of robots is tested multiple times and the priority rules were complied with in all tested situations.	Yes
4	Testing shortest path following behaviour	No robots should be visible outside the coloured paths that mark the shortest paths for all robots.	No unexpected diversions from indicated shortest paths are detected, indicating all robots follow their shortest paths.	Yes

The verification tests performed do not entirely complete the model verification. As there is an infinite number of input parameter combinations and variations possible, the verification of this agent-based model is never complete. By performing multiple different tests for the four different main parts of verifying agent-based models as proposed by Dam et al. (2010), an effort is made to gain a sufficient confidence in the developed simulation model.

4.7.2 Model Validation

To check whether the *right thing* is build, the developed model needs to be validated. Model validation is a process to check if a simulation model possesses a satisfactory range of accuracy that is consistent with the intended application of the model (Sargent, 2010). Simulation models are usually developed for a specific purpose. In this research the simulation model is developed to test whether or not autonomous transport robots are a feasible substitution of conveyor belts in the sorting process in a baggage handling system at medium-sized regional airports that operate in a point-to-point network. The validity of the model is determined with respect to this purpose. However, traditional validation methods are not always suitable for validating agent-based models. The agent-based model of the baggage robot concept cannot be validated by comparing computed behaviour to 'real' system behaviour, as there is no 'real' system available yet that can be used for comparison. The outcome of this simulation model rather serves as an increased insight and knowledge into the use of the baggage handling concept in the sorting process of baggage handling systems. To validate this outcome, several different methods are available, amongst which are (Dam et al., 2010):

- Historic replay
- Face validation through expert consultation
- Literature validation
- Model replication

Historic replay, meaning comparing a model to a real-world situation, can only be done for situations that exist in the real world. As the baggage robot concept does not exist in the real world in this form yet, this method cannot be used to validate the developed simulation model. Literature validation is only possible if other models, not being agent-based models, of the same or similar systems exist. There is a lot of literature on automated guided vehicles, but the application of more intelligent autonomous transport robots is still lacking. This is why literature validation also can't be used to validate the simulation model of the sorting process in the baggage robot concept. Model replication could be a way to validate the model of sorting process in the baggage robot concept, but is very labour intensive. This method requires – in an ideal situation – a research team that has not been involved in the development of the first model, to create a second model with a different system decomposition or with a different modelling technique. The outcomes of the two models can be compared to validate the first model (Dam et al., 2010). Due to time and resources constraints, this method is not preferred for this research. This leaves one validation method to validate the model of sorting process in the baggage robot concept: face validation through expert consultation.

Face validation is the most commonly used validation method for agent-based models. In this method, domain experts are interviewed to discuss the model and the application of the model for its designed purpose (Dam et al., 2010). Three different experts are interviewed to validate the developed simulation model. The first is a full professor and head of the school of aviation of the University of New South Wales in Sydney, Australia. He is an expert in transport systems and is specialized towards baggage handling systems, both the conventional baggage handling systems as well as the latest designs complying with IATA resolution 753. The second expert is also a full professor and the department head of the Delft Center for Systems and Control, a department of the Delft University of Technology. He is an expert in the field of high-tech vehicle control and has experience with automated guided vehicle systems in different industries. The third expert is a senior systems engineer at Vanderlande. He is an expert in high level control solutions, meaning the IT systems that use baggage and system information to match with business rules and determine the next step in the process.

The experts were provided with information on the baggage robot concept in general and the scope and assumptions of the developed simulation model. They were asked to answer several questions on baggage handling systems in general, individual transport robots in general and the combination of both into the baggage robot concept. The full questionnaire is provided in Appendix J. The experts face validated the developed model with respect to the model purpose. They acknowledged that conventional baggage handling systems have the main disadvantage that future extensions have to be taken into account from day one as especially sorter equipment is not easy to modify when it is in live operation. They argue that replacing these conveyor belts by individual transport robots can provide more flexibility.

The experts do mention that as a one to one comparison with conventional conveyor belt systems is not included it can't be proved whether or not the baggage robot concept is better than the conventional baggage handling system. To investigate the full potential of the baggage robot concept, several additions can be made to the model amongst which are:

- Including the security screening process
- An alternative version of the model with conveyor belts
- An alternative version of the model with guided paths that robots can follow, like an AGV system

However, the purpose of the simulation model is to provide an increased insight and knowledge into the use of autonomous transport robots in the sorting process of the baggage handling system. The experts believe that the level of detail of the model is sufficient for this purpose.

The developed simulation model is face validated by three different experts. Using the face validation through expert consultation as a validation method however, has some limitations. As it is a new concept that has not been thoroughly researched yet, expert might have a good understanding of what happens in conventional baggage handling systems, but may not have a very systematic understanding on what may happen to the baggage handling system when autonomous transport robots are used. Naturally, experts rely on their own idea of system behaviour to estimate model results. These ideas are all subjective and can be biased and flawed in various degrees (Dam et al., 2010). This is why models that are validated through the face validation method have face validity, meaning the model appears reasonable and that it looks like it will do what it is supposed to do. Although experts can be wrong too, Dam et al. (2010) argue that this method is still an appropriate way to address agent-based model validation, meaning that a model that is face validated by experts can be considered good enough.

4.8 Conclusion: Towards Model Experimentation

In this chapter, a simulation model of the sorting process in the baggage robot concept is developed using agent-based modelling. During the development of the model the insight and knowledge on the baggage robot concept in general is increased iteratively. The model is verified and after validation by experts the model is ready to be used for experimentation. Having verified and face validated the model does not mean it is 100% perfect. Further evaluation of strong and weak points of the model and important design parameters have to follow from structured experiments performed with the model under different scenarios. The next chapter is concerned with the model experimentation.

5. Model Experimentation

After the simulation model is created, verified and face validated, it can be used to perform experiments. With agent-based models, simulation experiments can be conducted. The real system can be modelled and experimented with to get an understanding of its behaviour and its performance. By performing experiments, various strategies for the operation of the system can also be evaluated (Siebers et al., 2010). In order to do so, experimental designs must be created. Aspects of the experimental design are a hypothesis and a time frame. The first step is testing the performance of the system at presumed normal conditions when it uses individual transport robots. The second step is to investigate the impacts of different situations – meaning that different values for input parameters to the system are used. Finally, based on the results of these two experiments, a third experiment can be executed to further investigate the impact of the two most important input parameters.

5.1. Hypothesis

According to Dam et al. (2010), there are two main types of hypothesis that can be formulated for agent-based models. The first type states that under specified conditions, a macroscopic regularity will emerge from the agent-based model. This hypothesis type is most common to use on models that try to explain real-world observed regularity by modelling a phenomenon that is expected to cause the regularity. Experimenting with the model provides an answer to the question whether or not the cause of the regularity is explained. If the model does not produce the expected regularity, the cause of the regularity can't be explained by the model and therefore the hypothesis is falsified. What this type of hypothesis also does is asking questions about the conditions that are needed to produce the real-world observed regularity. The types of models suitable for this first type of hypotheses try to provide an explanation of how the system works under varying circumstances and input values (Dam et al., 2010).

The second type of hypothesis states that 'a range of clearly identifiable emergent behaviours and regularities can be established from the agent based model of a system'. This type of hypothesis is not intended for models that attempt to recreate the real world but rather is used for exploration, as it tries to explain if and if so which parameters delay, alter or disrupt the regularity. This type of hypothesis is falsified when the expected regularity does not behave as desired, or if it only behaves as desired under unreasonable conditions, or when it behaves chaotically. Chaotic behaviour is when behaviour emerges that has no relation to the parameters investigated whatsoever. This type of hypothesis is mostly used when the modeller has no clear idea on what he or she is looking for in the model (Dam et al., 2010).

In the case of incorporating autonomous and individual transport robots in a baggage handling system at a medium-sized airport operating in a point-to-point network, the first hypothesis type is considered most relevant. The set of parameter values for this system is fairly clear as they are (loosely) based on real-world parameters. Next to having an idea on the values for the input parameters, the output metrics to collect are also clear, based on the KPIs for the baggage robot concept described in section 3.4.4. The experiments on the simulation model of the sorting process in the baggage robot concept therefore focus on examining these output metrics over time, to clarify if regularity can be discovered and when. Next to that, experiments can also show whether the regularity is stable or not.

In contrast to the second type of hypotheses, the first type will most likely have a fairly clear set of parameter values since they are based on real-world parameters that match the system of interest. By choosing this type of hypothesis, experiments that can be executed by the simulation model focus on examining the variation of the input parameters over time to clarify differences in the performance of the system (Dam et al., 2010).

Considering the topic and scope of this research, the question is which combination of parameter values is beneficial for the performance of the sorting process of a baggage handling system. The hypothesis is that under specified conditions, a macroscopic regularity will emerge from the agent based model. The specified conditions are the input parameters and the macroscopic regularity is that the values of the output metrics follow a certain regularity. For example: the more robots present in the system (input parameter), the lower the average process time of bags (output metric). Of course the number of robots present is not the only input parameter influencing the average bag process time. The type of layout or configuration is also expected to have (limited) influence, as well as the values for the battery reduction and charging rates. For example, the lower the charging rate, the faster robots can charge and as a consequence they can spend more time transporting bags. To accept the hypothesis, the model needs to produce the desired regularity – in this research, performance, measured in KPIs – otherwise the hypothesis is falsified (Dam et al., 2010).

To test the hypothesis and to answer the question on the performance of the system in case individual transport robots are incorporated, the output metrics should include the key performance indicators that have been formulated in section 3.4.4.

To experiment with the simulation model, the characteristics of the robots (e.g. number of robots present, battery power reduction and charging rates) can be varied, as well as other variables that influence the possibilities of the system (e.g. the number of bags entering the system on a daily basis and the configuration or layout of the floor space). A combination of these input parameters is also possible. Experiments try to obtain results for output metrics, including the KPIs, by varying input parameter values.

5.2. Time

In agent-based models there is a strong focus on observing behaviour in a system. In this research the behaviour is the varying level of performance that is achieved by the system. When the focus is on behaviour, it is hard to predict upfront how much time it takes for certain behaviour to arise. This makes it hard to determine upfront the required number of iterations. In models that focus on the second type of hypothesis, emergent behaviour can be unstable or oscillatory and can be sensitive to initial conditions. The first type of hypothesis that is used in this research has the advantage that some real world conditions can be replicated, providing a real-world timeframe (Dam et al., 2010).

In the case of incorporating autonomous and individual transport robots in a baggage handling system at a medium-sized regional airport, it is most interesting to define timeframes that are detailed to seconds. The system state can change from fully operational to a total breakdown in a matter of seconds. Taking larger time steps is therefore undesirable. When it comes to the timeframe, baggage handling systems operate in daily cycles. The system is not operational 24 hours a day since night flights are not allowed to depart or land. This implies that the baggage handling system is idle during the night. To get a grasp on the use of the transport robots in baggage handling systems, a day is therefore considered sufficient to capture typical daily demand patterns. This can be extrapolated to seasonal and yearly patterns by changing the value of input parameters like the arrival pattern of bags.

5.3. Experiment Setup

For this research, three experiments are set up. Based on Dam et al. (2010), the first two experiments are constructed. The third experiment is set up by the researcher. The first experiment is an attempt in representing the most ideal practical situation, in which the input parameter values are set to represent 'normal conditions'. Normal conditions are considered to be the conditions on an average day in the year with average external conditions – so no unexpected disruptions like dysfunctional systems or heavy snowfall. By fixing the parameters that represent normal conditions, the performance of the baggage robot concept in the sorting part of a baggage handling system can be examined. This 'base case' will be run with 50 replications. These input parameter values are based on real data for a medium-sized regional airport in the Netherlands and assumptions on the battery quality.

The second experiment however requires variation in the input parameters values but within the practical limitations of the model. These values need to be established to be able to run multiple different experiments in a smart way. For designing the experiment input parameter values, different techniques can be used. A widely-used technique in the probabilistic analysis of engineering systems is Monte Carlo simulation. This technique performs numerical experiments with the goal of obtaining the statistics of the output variables of a system model. Figure 26 shows a rough visualization on how this technique works. The statistics of the input variables (X1, X2, .. , Xk) are given and can vary – in the figure all variables have a normal distribution. These distributions are used to obtain the statistics of the output variable Y. The black box in the figure represents the computational model that is used to transform the input variables to output. Based on their distributions, the values of the input variables are sampled in each experiment. Multiple experiments are done in the same way, resulting in computed statistics of the output variable(s) (Cruse, 1997).

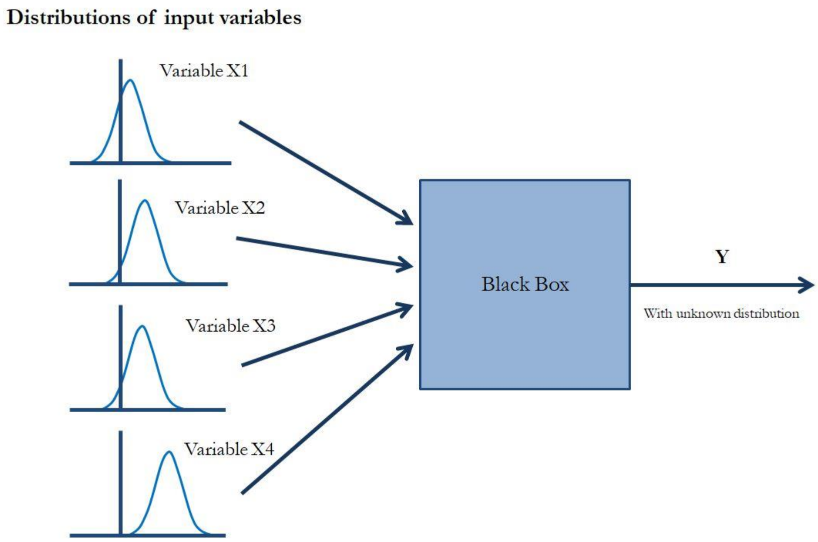

Figure 26 - Experiment technique (Image Author, 2017)

Monte Carlo simulation is mostly used for complex problems, since the method is simple, adequate and widely employed. However, for analysing the reliability of complex engineering systems, Monte Carlo simulation is very time consuming and expensive in terms of computer resources. This opens up possibilities for other methods, such as Latin Hypercube Sampling (LHS) (Olsson, Sandberg, & Dahlblom, 2003). Latin Hypercube Sampling supports Monte Carlo simulation by generating samples of the input variables. By doing so, the estimates become more precise (Wyss & Jorgensen, 1998). To quote Dam et al. (2010):

"Latin Hypercube Sampling is a statistical technique that guarantees uniform sampling with the desired granularity of the scenario space given a Y dimensional parameter space and with a limit of X experiments."

Latin Hypercube Sampling considers the entire set of parameters, instead of considering only one parameter at a time, which is the case with random parameters. It finds where in the parameter space the predetermined number of experiments should be performed to get the most representative subset of the space. In this research, the LHS function in the software tool R is used to generate LHS samples.

As the second experiment contains an exploration on how the input parameters affect the performance of the baggage robots in the sorting process of a baggage handling system, a smart choice of experiments or 'runs' need to be made. To select a subset of experiments over the wide-ranging parameters, Latin Hypercube Sampling will be used. This experiment tries to indicate which input parameters affect the output metrics the most.

The third experiment investigates the most influential input parameters further. By only varying the number of robots and the layout of the manoeuvring area of the robots, while keeping the other input parameters constant, the impact of changes in these two input parameters can be demonstrated with more certainty.

5.3.1. Replications

According to Dam et al. (2010), the ideal number of replications for an agent-based model is 100 replications per combination of input parameter values. This number of replications is not feasible in this research, given the constraints in time and available simulation resources. The three experiment setups briefly described in the previous section have all been run with varying number of replications. The first experiment setup uses 50 replications, the second setup uses 30 and the third setup is only replicated once for the different combinations of input parameter values.

The first experiment setup approximates the ideal number of replications. The second experiment setup is at the lower boundary of acceptability in terms of number of replications. The last experiment is simply not suitable for any replications due to the large number of parameter combinations already included. This limits the possibility to stating definitive conclusions on the statistical significance of the results of this experiment setup. Further simulation experiments and analyses of results should allow for detailed comparison of the deviation induced by replicating these input parameters relative to the deviation induced by changing input parameter values.

5.4. Parameters

The simulation model has eight different input parameters that can be altered to generate results. For each of the parameters a range is defined in which the parameter value can vary. The input parameters, their value range and the step size within these ranges are shown in Table 10.

The number of bags that enter the sorting process of the baggage handling system during the day according to a normal distribution are set to be either 1624 or 4290. The value 1624 corresponds to the average number of bags handled by the baggage handling system of a medium-sized regional airport in the Netherlands. These 1624 bags are the bag load of 13 flights that are distributed over the day. These flights with accompanying destinations, aircraft types, departure times and first and last check-in times can be found in Appendix F. The other value of 4290 is based on the same number of flights at that same airport, but assumes that all flights are executed by one of the largest passenger aircraft, increasing the number of passengers and therefore bags on a day by 164% to 4290. The storage and charging area configurations are discussed in section 4.4.4.2. and the input parameters that concern the charging and battery power consumption rates of robots under different circumstances have been discussed in section 4.4.4.3.

Table 10 - Parameter value variations for experiment designs

Input Parameter	Value Range	Step size
Number of bags entering during the day according to a normal distribution	1624 or 4290	Not applicable
Storage and charging area configuration	Central areas Six Vertical Areas Four Horizontal Areas Six Horizontal Areas	Not applicable
Number of transport robots	1-96	1
Battery power reduction rate when idle	0 – 0.01 %/sec	0.0001
Battery power reduction rate when driving without a load	0 – 0.01 %/sec	0.0001
Battery power reduction rate when driving with a load	0 – 0.01 %/sec	0.0001
Battery charging rate	0 – 0.01 %/sec	0.0001
Battery level threshold	0 – 100 %	0.1

5.5. Experiment Design I – Fixed Settings

In the first experiment design, all settings are fixed to focus on the performance of the baggage handling system when autonomous and individual transport robots are used to perform the sorting task of the baggage handling system. The key goal of this experiment is to get to know the order of magnitude of the KPIs of the system under presumed normal conditions. This 'base case' is based on indicated assumptions.

General setup

- Ticks [50.700]: Each tick = 1 second. 50.700 ticks represent 14 hours and 5 minutes. This simulates a busy day at a medium-sized regional airport in the Netherlands with 13 flights departing between 07:10 and 19:05. The simulation starts at 05:00 when the baggage handling system opens. The first bags can be checked in from 05:10, two hours before the first flight departs. The final flight departs at 19:05, which is 14 hours and 5 minutes later than 05:00. This tick limit therefore considers the operational time of one day. A day is considered long enough to explore the performance of the system under normal conditions.
- Repetitions [50]: Sufficient to get a good sample of results when the settings are fixed.

Parameter settings

- Total number of bags arriving according to an arrival pattern [1624]: A busy day at a medium-sized regional airport in the Netherlands is used as input data. Appendix F shows the flight numbers, destinations, check-in times and departure times, as well as the aircraft types per flight. Based on these aircraft types the maximum number of passengers is derived, which in this experiment are adjusted considering the passenger load factor for 2017 which is 80.7%. The arrival pattern is considered to be normally distributed with a mean of 40 minutes and a standard deviation of 30 minutes. The distribution is cut off at 0 and 80 minutes representing the first and last check-in time.
- Storage and charging area configuration [six horizontal areas]: This layout uses the smallest floor space out of all four configuration options. A small floor space increases the risk of collisions as robots have less space to divert efficiently. This layout is more usable in smaller baggage handling areas than the other configurations
- Number of transport robots [8]: Based on experiments ran by Vanderlande at a regional airport in the Netherlands

- Battery power reduction rate when idle [-0.00138889]: As explained in section 4.3.3.3.
- Battery power reduction rate when driving without a load [-0.00416667]: As explained in section 4.3.3.3.
- Battery power reduction rate when driving with a load [-0.0055556]: As explained in section 4.3.3.3.
- Battery charging rate [+0.00694444]: As explained in section 4.3.3.3.
- Battery level threshold [0.8]: As explained in section 4.3.3.3.

Output at each tick

- The tick
- The cumulative number of bags in the system
- The cumulative number of bags leaving the system
- The average time a bag spends:
    - On an incoming conveyor belt
    - Waiting for a robot to get assigned to the bag
    - Waiting for the assigned robot to arrive at the incoming conveyor belt
    - On a robot
- The average total process time of bags
- The average total trip time of robots
- Percentage of bags that don't make it to one of the makeup stations in time
- Percentage loaded trips by robots
- Percentage empty trips by robots
- Percentage charging time of robots
- Total number of potential conflicts detected and avoided

5.6. Experiment Design II – Parameter Sweep Experiment

For the second experiment – a parameter sweep experiment – a subset of experiments is selected using Latin Hypercube Sampling for the eight input parameters. In the first experimental design, the number of bags arriving following a normal distribution was considered constant, having a value of 1624. For the parameter sweep experiment, an additional and extreme case is added to create more variety. As the scope doesn't change, the same number of flights (13) is used but the aircraft type for all flights is now assumed to be a Boeing 777, the world's largest twinjet. The Dutch flag carrier KLM operates this type of aircraft with a seat capacity of 330. For the variety in the parameter sweep experiment, no correction for the passenger load factor is applied in this experiment. The assumption is therefore that all seats are occupied and each passenger takes one bag, resulting in 13 times 330 bags to be distributed over the 13 flights. A histogram of this extreme value for the bag arrival pattern can be found in Appendix G.

The general setup for experiment design II is the same as for experiment design I, being 50.700 ticks per run. Each combination of input parameter values is replicated 30 times. The type of output at each tick is also the same as in the first experiment design. Due to limitations in computer power, thirty samples of combinations of input parameter values are drawn. The parameter settings however differ. Latin Hypercube Sampling has provided thirty unique combinations of parameter setting values. The ranges in which the different input parameter values vary are displayed in Table 11. The complete list of all the combinations tested is documented in Appendix H.

Table 11 - Parameter Value Ranges in the Parameter Sweep Experiment

1624 Bags		4290 Bags	
Number of Robots	4 - 93	**Number of Robots**	2 – 91
Configuration	Central Areas Four Horizontal Areas Six Horizontal Areas Six Vertical Areas	**Configuration**	Central Areas Four Horizontal Areas Six Vertical Areas
Battery reduction rate idle [%/sec]	0.00032 - 0.00941	**Battery reduction rate idle [%/sec]**	0.00119 – 0.0097
Battery reduction rate empty [%/sec]	0.00028 - 0.00998	**Battery reduction rate empty [%/sec]**	0.00075 – 0.0088
Battery reduction rate loaded [%/sec]	0.0003 - 0.00892	**Battery reduction rate loaded [%/sec]**	0.00089 – 0.00977
Charging rate [%/sec]	0.00022 - 0.00838	**Charging rate [%/sec]**	0.00066 – 0.00973
Battery level threshold [%]	3 - 93.4	**Battery level threshold [%]**	5 – 97.6

5.7. Experiment Design III – Influencing Parameters Effect

For the second experiment – a parameter sweep experiment – a subset of experiments is selected using Latin Hypercube Sampling for the eight input parameters. In the first experimental design, the number of bags arriving following a normal distribution was considered constant, having a value of 1624. For the parameter sweep experiment, an additional and extreme case is added to create more variety. As the scope doesn't change, the same number of flights (13) is used but the aircraft type for all flights is now assumed to be a Boeing 777, the world's largest twinjet. The Dutch flag carrier KLM operates this type of aircraft with a seat capacity of 330. For the variety in the parameter sweep experiment, no correction for the passenger load factor is applied in this experiment. The assumption is therefore that all seats are occupied and each passenger takes one bag, resulting in 13 times 330 bags to be distributed over the 13 flights. A histogram of this extreme value for the bag arrival pattern can be found in Appendix G.

In this experiment, the number of robots varies for the two scenarios as follows:

1624 Bags		4290 Bags	
Number of Robots	1-25	**Number of Robots**	1-35

The input parameters related to the floor layout configuration and the robot's battery remain constant for both scenarios:

- Configuration: Central Areas, Four Horizontal Areas, Six Horizontal Areas, Six Vertical Areas
- Battery reduction rate idle [%/sec] 0.00138889
- Battery reduction rate empty [%/sec] 0.00416667
- Battery reduction rate loaded [%/sec] 0.0055556
- Charging rate [%/sec] 0.00694444
- Battery level threshold [%] 0.8

5.8. Results from Experiments

The experiments described in the previous sections are executed with the developed simulation model. In this section the results of the three experiments are discussed. For each experiment, the most important output metrics are given. These metrics are the seven previously described KPIs for the baggage robot concept.

1. Average process time of bags
2. Minimum number of robots
3. Percentage empty robot trips
4. Percentage loaded robot trips
5. Percentage charging time
6. Number of avoided conflicts
7. Percentage of mishandled bags

Next to these seven KPIs, an additional output metric is calculated in every experiment. This metric shows the number of bags handled during runtime. As described, the runtime for each experiment is the same, being 50.700 seconds. If the value for this metric is lower than 1624 for the 1624 scenario or lower than 4290 for the 4290 scenario, it means that the specific combination of input parameter values create a system that is unable to handle all the bags. This results in an unacceptable value for mishandled bags and therefore these input parameter value combinations are considered infeasible. Experiment designs two and three are performed twice; once for the 1624 bags scenario and once for the 4290 scenario.

5.7.1 Fixed Settings Experiment

In the fixed settings experiment, all input parameters have a fixed value as described in section 5.5. The experiment is performed with 50 replications and gives the results for the seven KPI output metrics and the additional handled bags output metric. For every replication, the 'handled bags in runtime' metric gives a value of 1624, so the combination of input parameter values result in a system that is capable of handling all 1624 bags. For each KPI a theoretical optimal value is determined (as explained in section 3.4.4) and these are shown in the table below.

Table 12 shows the obtained optimal value of the seven KPIs for the fixed settings experiment. For each obtained optimal value, four descriptive statistics are added to show the deviations for the KPIs in the 50 replications. These descriptive statistics do not apply to the second KPI, being the number of robots, as this is both an input parameter.

Table 12 - Results from fixed settings experiment with 50 replications for 7 KPIs

KPI	Theoretic Optimal Value	Obtained Optimal Value	Min	Max	Average	Median	Standard Deviation
1	≤ 6 minutes or 360 sec	136.96 sec	136.96 sec	154.44 sec	146.84 sec	147.47 sec	3.18 sec
2	As small as possible	8	*Not applicable*				
3	Close to 50%	51.92%	51.92%	52.35%	52.14%	52.15%	0.09%
4	As high as possible	48.08%	47.65%	48.08%	47.86%	47.85%	0.09%
5	As low as possible	26.68%	26.68%	27.03%	26.86%	26.85%	0.08%
6	As low as possible	976.00	976.00	1164.00	1071.57	1064.00	44.82
7	Between 0 and 0.02%	0.43%	0.43%	2.77%	1.59%	1.60%	0.51%

To visualize the distributions of several output metrics over 50 replications, Figure 27 shows four boxplots of four KPIs:

- the average process time of bags,
- the percentage charging time,
- the number of avoided conflicts
- the percentage of mishandled bags.

The line within the boxplot indicates the median, the ends of the boxes indicate the upper (third) and lower (first) quartile and the whiskers show the minima and maxima. The dots represent possible outliers.

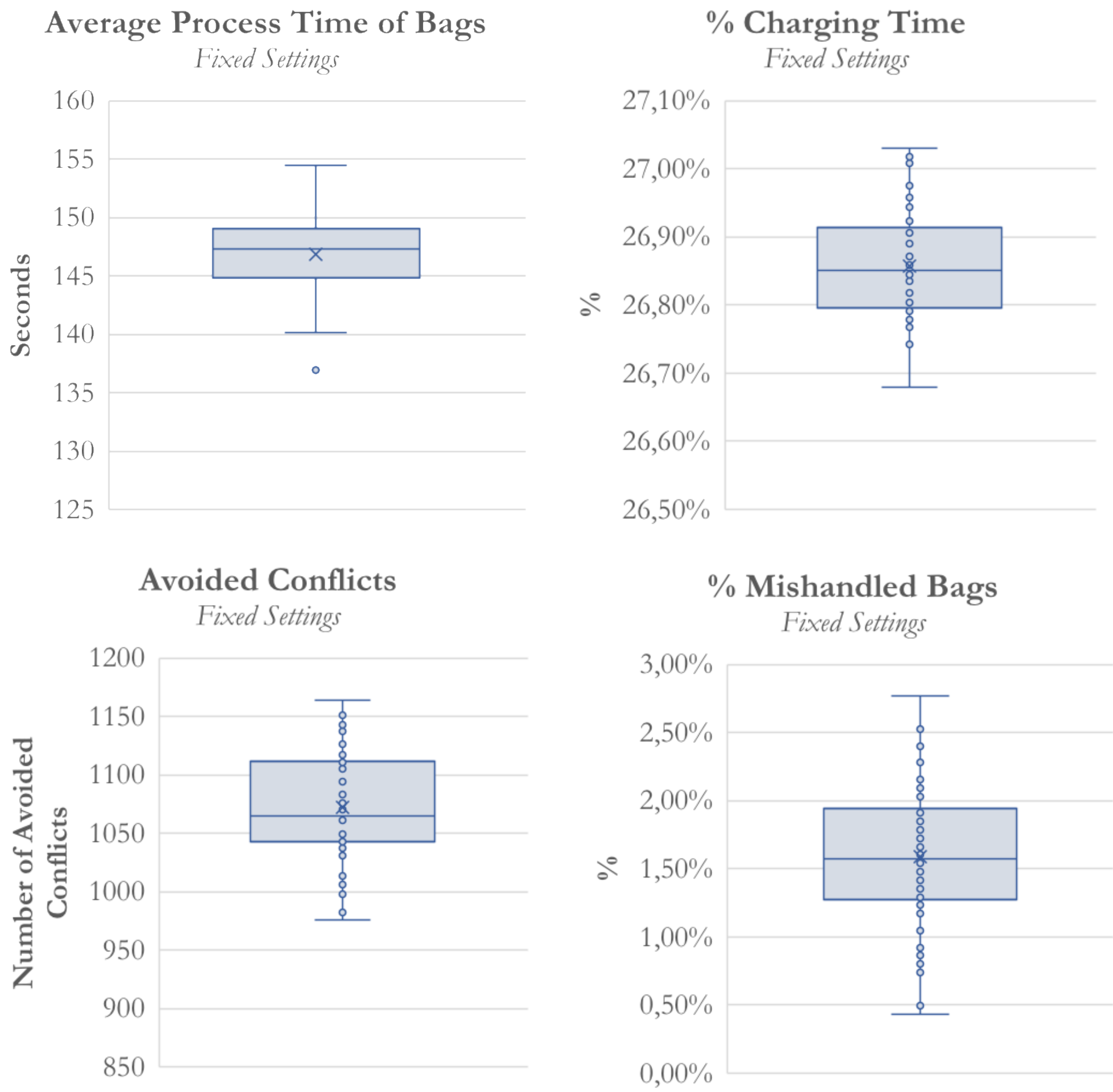

Figure 27 - Boxplots of Fixed Settings Experiment Results for 50 replications

Table shows that the minimum and average process time of bags in the fixed settings experiment is lower than six minutes. This means the combination of input parameter settings complies with the set theoretical optimal value. In this experiment, the robots are very close in complying with the occupancy rate, the 50%-50% ratio between the percentage empty and loaded trips. The average percentage of charging time over the 50 replications is almost 27%. The number of avoided conflicts is 976. This value however does not necessarily mean that 976 unique bags were on robots that were in a conflict. It is possible that a bag was transported by a robot that was in a longer lasting conflict, adding to this metric every second. The percentage of mishandled bags has a minimal and optimal value of 0.43%, corresponding to 7 mishandled bags during the simulated day.

Interpretation of results

The average process time of bags is far below the maximum of 6 minutes or 360 seconds. During the experiment one of the replications had a maximum value for this KPI of 154.44 seconds, which is still way below the 6 minutes threshold. Even when unexpected minor disruptions become present, there is enough slack time to avoid significant problems. This combination of input parameter values is able to meet the average process time of bags KPI. A 50%-50% ratio for the utilization rate of robots is considered ideal when the locations of the charging and storage areas do not cause any detours for the robots. However, this is not the case as the layout configuration chosen in this experiment can cause a minor detour for the robots as they can choose to which charging and storage position they want to go. A value that comes close to this 50%-50% ratio is therefore considered optimal. This experiment shows a ratio of 51.92%-48.08% for empty and loaded trips, which is considered within the ranges of being optimal.

Considering the percentage of charging time, it should be as low as possible while ensuring all bags are handled, which is the case in this experiment. A high percentage of charging time might mean there are too few robots in the system meaning the robots that are present each have to perform multiple transportation tasks in a row, reducing their battery power, which causes a need for charging. A high percentage of charging time however might mean there are too many robots in the system. This can mean there are robots in the system that do not perform a transportation task at all, so their battery level is not reducing and there is no need to charge. When robot batteries are not emptying because they are not moving, they simply remain available while standing on a charging area, but this time spent does not count as charging time. A value of 26.68% for 8 robots is considered fine, indicating a good balance between the number of robots and the number of incoming bags over time. The number of avoided conflicts cannot be judged to be low or high. This combination of input parameter values leads to a situation in which the percentage of mishandled bags is higher than 0.02% in all runs. The minimum value obtained in the experiment does not comply with the set maximum of 0.02% and the maximum value is even larger, 2.77%. This means the selected set of input parameters is not able to comply with all the theoretic optimal values for the KPIs, showing that the base case values chosen do not fulfil the practical requirements to the baggage robot concept. The next experiment shall provide more insight into which combination of design parameter values actually provides a feasible design.

The whiskers show the order of magnitude of variation in the KPI values for different simulation seeds. Different seeds mean that for each replication different and random choices in for example conflict resolutions are made. Through the path dependency in the simulation, a slightly different situation emerges at the end of the simulation run. By showing the spread in values that occur – of which the extremes are displayed by the whiskers – the implication of these random choices made in the different and random seeds are explored.

5.7.2 Parameter Sweep Experiment

In the parameter sweep experiment, all input parameters have been varied for both the 1624 bags and the 4290 bags scenario as indicated in section 5.6. The experiment is performed with 30 replications of each unique combination and gives the results for the seven KPI output metrics and the additional handled bags output metric. Not for every combination of input parameter values the 'handled bags in runtime' metrics gives a value of 1624 for the 1624 scenario or 4290 for the 4290 bags scenario. Appendix K.1. shows the infeasible combinations for both scenarios, as well as the average process time of bags (APT), the average percentage of mishandled bags (AMH), the average percentage of loaded trips (ALT), the average percentage of empty trips

(AET) and the average percentage of charging time (ACT). The final column in the table in Appendix K.1. shows some descriptive statistics on the number of bags that were handled. The inability of these five different combinations of input parameter values to handle all the bags that are inserted in the system in the simulated day, indicates that the baggage robot concept is not able to perform the sorting process under all simulated input values.

1624 Bags Scenario

Table 13 shows the obtained optimal value for the seven KPIs for the parameter sweep experiment for the 1624 bags scenario. As in the fixed settings experiment, four descriptive statistics are added to show the deviations for the KPIs. All the combinations that resulted from the Latin Hypercube Sampling are shown in Appendix H. All these combinations were replicated 30 times. As mentioned earlier, multiple combinations were found to be unable to handle the required number of 1692 bags within runtime. These combinations can be found in Appendix K.1. These infeasible combinations are still included in Table 13. However, when excluding these infeasible input parameter combinations from the analysis, different results for the KPIs are obtained.

In Table 13 red cells represent the cells that contain values for the KPIs that were obtained from an infeasible combination of input parameter values, i.e. combinations that are unable to handle the 1624 bags within the runtime.

The transparent cells show the values when only feasible combinations are considered. The combination of input parameter values that result in the values for the obtained optimal values of the KPIs in Table 13 can be found in Appendix K.2. As mentioned in the parameter sweep experiment, the descriptive statistics do not apply to the number of robots KPI as it is also used as an input parameter.

Table 13 - Results from parameter sweep experiment 1 with 30 replications for 15 settings and for 7 KPIs

KPI	Theoretic Optimal Value	Obtained Optimal Value	Min	Max	Average	Median	Standard Deviation
1	≤ 6 minutes or 360 sec	61.75 sec	61.75	4492.15	366.28	68.29	1062.67
				94.80	69.81	68.00	9.89
2	As small as possible	15	*Not applicable*				
3	Close to 50%	50.94%	50.94%	58.27%	54.24%	54.71%	2.29%
				57.81%	53.86%	54.65%	2.18%
4	As high as possible	49.06%	41.73%	49.06%	45.76%	45.29%	2.29%
			42.19%		46.14%	45.35%	2.18%
5	As low as possible	0.14%	0.14%	77.95%	17.01%	5.80%	23.17%
				22.41%	8.34%	5.70%	7.33%
6	As low as possible	170.00	170.00	1873	1470.94	1584.50	371.93
		1126.00	1126.00	1873.00	1566.80	1608.00	169.83
7	Between 0 and 0.02%	0.00%	0.00%	76.22%	5.73%	0.00%	18.93%
				0.00%	0.00%		0.00%

Table 13 shows that the obtained optimal value for the average process time of bags is lower than 6 minutes in the parameter sweep experiment for 1624. However, when infeasible combinations are included, both the average and maximum value for this KPI exceeds the 6 minutes or 360 seconds threshold. When these infeasible combinations are excluded, the average and maximum value drops below the threshold again. The same holds for the last KPI, the percentage of mishandled bags. Other significant differences between the situations where infeasible combinations are included and situations where they are excluded can be seen in the KPIs ‘percentage charging time’ and ‘number of avoided conflicts’. The difference in the values for the percentage of empty and loaded trips is not too big.

To visualize the distributions of several output metrics over 30 replications, Figure 28 shows four boxplots of four KPIs:

- the average process time of bags
- the percentage charging time
- the number of avoided conflicts
- the percentage of mishandled bags.

These boxplots include both the feasible and the infeasible combinations of input parameter values. This way, the effect of infeasible input parameter value combinations is visible.

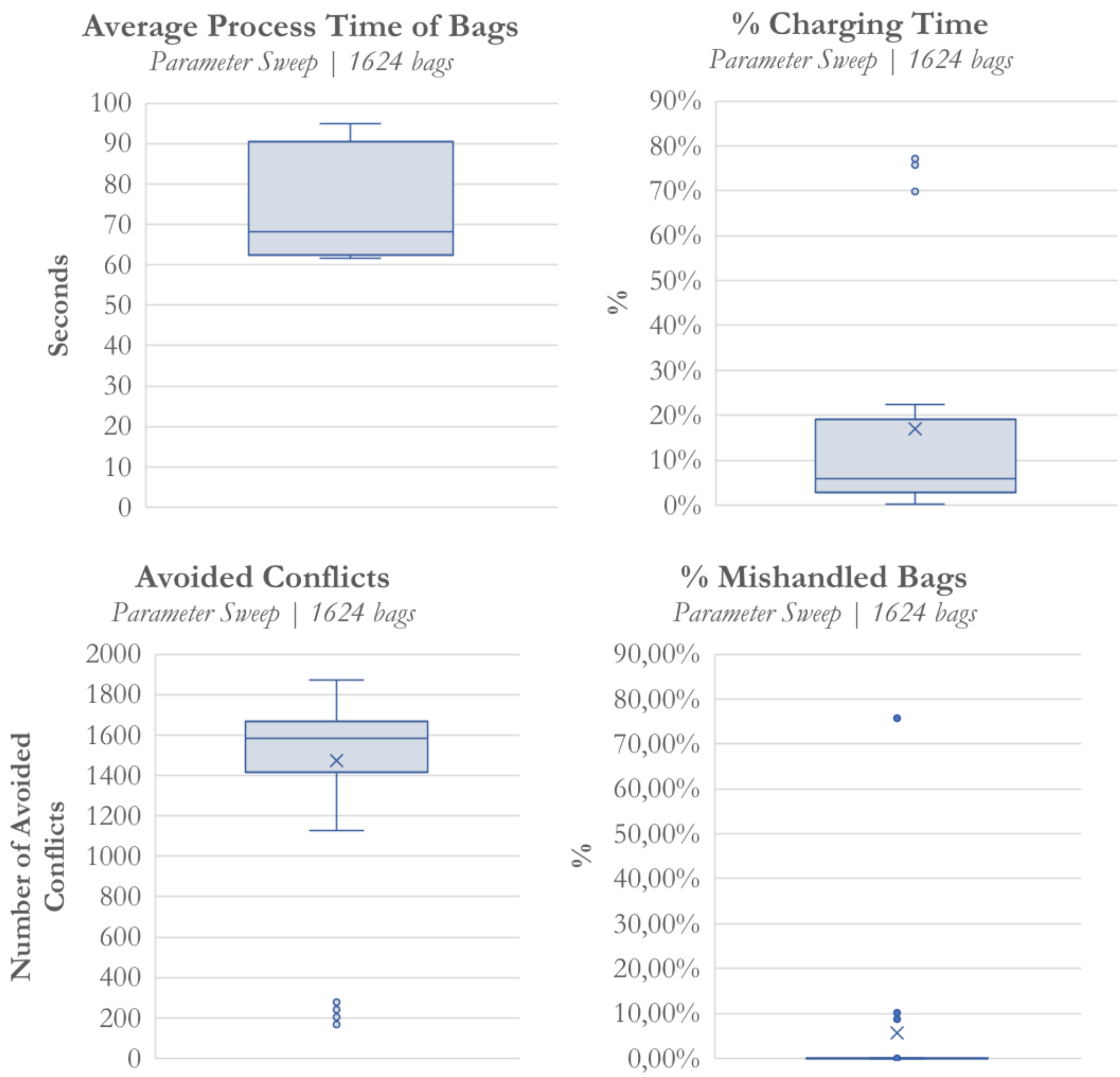

Figure 28 - Boxplots of Parameter Sweep Experiment with 1624 bags. Results for 30 replications.

The obtained optimal values for the KPIs in the parameter sweep experiment for 1624 are obtained by feasible combinations of input parameter values, except for the KPI ‘number of avoided conflicts’. For this KPI, a value of 170 is the minimal value obtained when both feasible and infeasible combinations are analysed. This value is the result of an infeasible combination, stated in the first row of the infeasible combinations table in Appendix K.1. This combination was the four horizontal areas layout with only four robots and a very low charging rate. As only four robots tried to handle all the bags – which they were unable to do; on average they were able to handle only 558 out of 1624 bags – the chances of them running into each other were lower, causing the possible number of conflicts to be very low.

Appendix K.2. shows the combinations of input parameter values that result in the obtained optimal values for the KPIs. What is most noticeable is that one specific combination of input parameter values causes the optimal value for three of the seven KPIs. This combination has 15 robots in the six horizontal areas configuration. The combination that has the lowest value on the average process time of bags combines 53 robots with the six horizontal areas configuration. 53 robots is not the highest number of robots present in the experiment. The 1624 bags scenario is also performed for 63, 66, 69, 73, 76, 82 and 93 robots in different layout configurations. As 53 robots show the lowest value for the average process time of bags, this implicates that if more than these 53 robots are present, they get in each other's way while moving around, resulting in more conflicts to be avoided, resulting in more time the robots have to wait for each other, resulting in a higher average process time for bags.

The lowest number for the avoided conflicts KPI is caused by 19 robots in the central areas configuration. For the 1624 bags scenario, there are only two combinations that have a lower number of robots. It is expected that the lower the number of robots is, the lower the value for the number of avoided conflicts is, as the size of the area remains constant so the chances of two or more robots running into each other is lower than when the area is occupied by a larger number of robots. The combinations in the 1624 bags scenario that have a lower number of robots are one of the infeasible combinations – four robots in the four horizontal areas configuration – and a configuration with 15 robots in the six horizontal areas layout. It would be in line of expectation if this second configuration would score better on the number of avoided conflicts KPI as it has 15 robots instead of 19 and is able to handle all the 1624 bags. However, this is not the case as the lowest value for number of avoided conflicts in the 15 robots in the six horizontal area configuration is 1297. This can be explained by the type of layout configuration. In the central areas configuration – combined with 19 robots resulting in the most optimal value for the KPI – there is more manoeuvring space for the robots than in the six horizontal areas configuration. This demonstrably results in less avoided conflicts.

The same experiment is also run for a more extreme scenario, a scenario in which 4290 bags need to be handled instead of 1624. The next section shows the result of these experiments.

4290 Bags Scenario

Table 14 shows the obtained optimal value for the seven KPIs for the parameter sweep experiment for the 4290 bags scenario. When it comes to the red coloured cells, the same reasoning applies as in the 1624 bags scenario – red cells represent the values for the KPIs and its descriptive statistics when infeasible combinations are taken into the data analysis as well. The infeasible combinations of input parameters for the 4290 bags scenario are visible in Appendix K.1. The combinations that do show the optimal values are visible in Appendix K.3.

All obtained values comply with the theoretic optimal value for the KPIs as stated. Compared to the 1624 scenario, the average process time of bags is 26.25% higher – still easily complying with the maximum threshold of 360 seconds – while the number of bags to be transported is 164.16% higher. The minimum number of robots necessary to handle all these 4290 bags is however higher, 26 compared to 15, having the same layout configuration. When 4290 bags need to be handled, the percentage of empty trips is slightly higher compared to the 1624 bags situation, which can be explained by the fact that the combination of input parameter values causing this optimal value for the percentage of empty trips includes having 57 robots. It is expected that at certain moments most of these robots are operational and as there are quite many, they might run into each other a lot. This results in more conflicts to be avoided which leads to more waiting time especially for empty robots, as loaded robots get priority over empty robots, increasing the percentage of empty trips in time. The same logic applies to the percentage of loaded trips. The optimal value for charging time is higher than in the 1624 bags scenario, which can be explained by the fact that the obtained optimal value for this KPI in the 1624 bags scenario is obtained by a combination of input parameter values that includes having 76 robots, while in the 4290 bags scenario this value is obtained by a combination of input parameter values that includes having 51 bags. A lower number of robots have to transfer significantly more bags, resulting in more need for charging and thus a higher percentage of charging time. The number of avoided conflicts is also significantly higher in the 4290 bags scenario compared to the 1624 bags scenario, due to the number of robots present in the combination that results in the optimal values; 19 robots for 1126 avoided conflicts in the 1624 bags scenario compared to 34 robots for 7733 avoided conflicts in the 4290 bags scenario.

Table 14 - Results from parameter sweep experiment 2 with 30 replications for 15 settings and for 7 KPIs

KPI	Theoretic Optimal Value	Obtained Optimal Value	Min	Max	Average	Median	Standard Deviation
1	≤ 6 minutes or 360 sec	77.96 sec	77.96	12871.93	1988.96	81.18	3891.19
				386.06	121.03	80.46	90.19
2	As small as possible	26	*Not applicable*				
3	Close to 50%	54.96%	54.96%	60.40%	56.36%	55.36%	1.61%
				57.71%	55.68%	55.29%	0.87%
4	As high as possible	45.04%	39.60%	45.04%	43.64%	44.64%	1.61%
			42.29%		44.32%	44.71%	0.87%
5	As low as possible	1.93%	1.93%	59.04%	26.54%	27.91%	16.55%
				46.53%	23.76%	26.76%	15.34%
6	As low as possible	69.00	69.00	9068.00	6515.21	8160.00	2971.58
		7733.00	5042.00	9068.00	7895.07	8276.00	1189.96
7	Between 0 and 0.02%	0.00%	0.00%	85.58%	17.10%	0.00%	30.53%
				15.24%	2.02%		4.57%

For the 4290 bags scenario in the parameter sweep experiment, boxplots of the average process time of bags, the percentage charging time, the number of avoided conflicts and the percentage of mishandled bags are shown below in Figure 29.

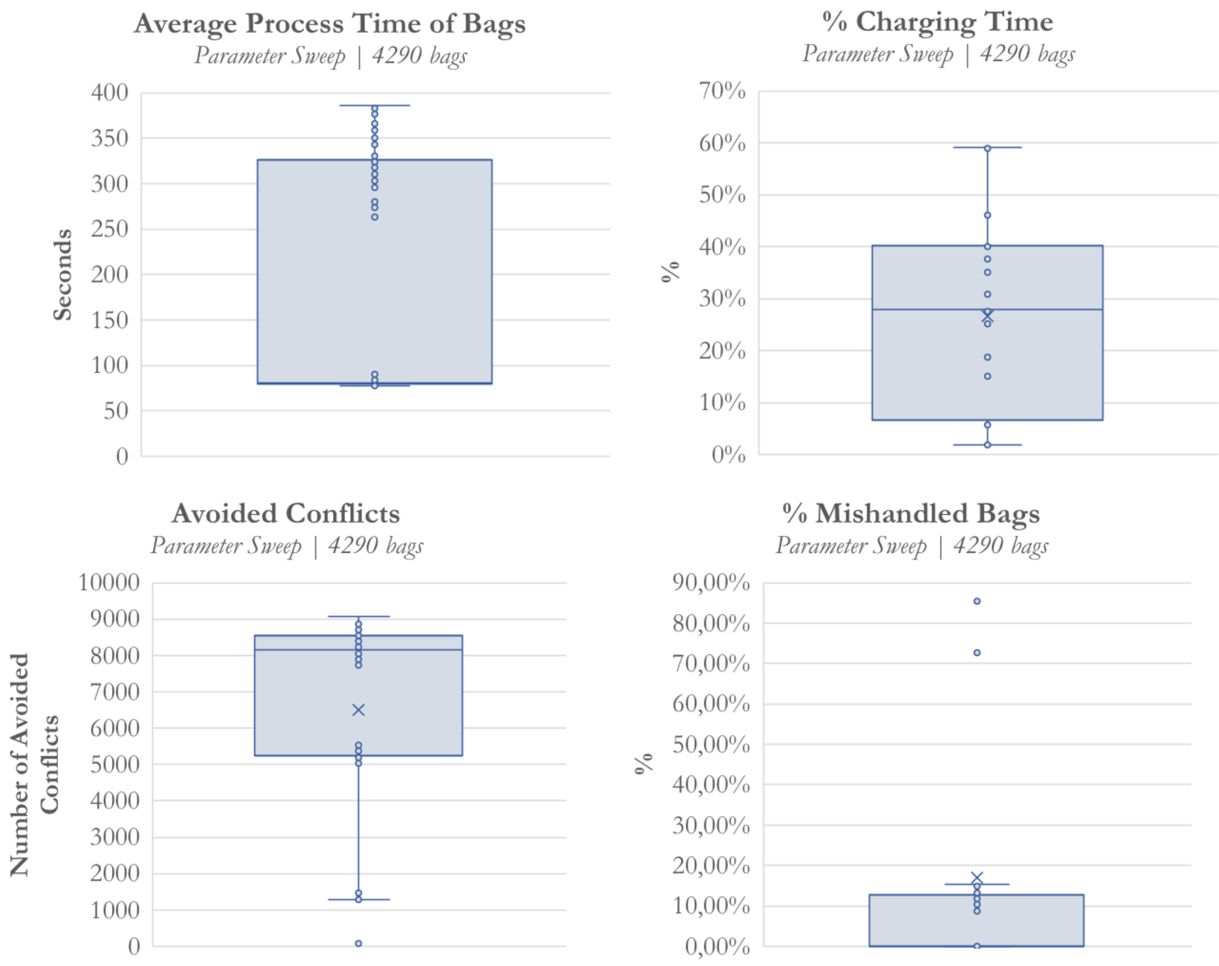

Figure 29 - Boxplots of Parameter Sweep Experiment with 4290 bags. Results for 30 replications

All the values for the obtained optimal value comply with the theoretic optimal values for the 4290 bags scenario. Compared to the 1624 bags scenario, the deviations in the KPI values for the 4290 bags scenario are larger, which is visible when the boxplots in Figure 28 and Figure 29 are compared.

Experience with the model shows that two input parameters are of most importance; the number of robots and the layout configuration. In the design process these two choices have to be made first as they are the key drivers for the required capital investment and operational expenditures. These factors are out of scope in this research, but cannot be neglected when assessing the performance of the scoped model.

The next section examines the effects of the input parameters 'number of robots' and 'configuration' more closely as they appear to have the largest impact on the KPIs. Both the 1624 and the 4290 bags scenario are examined for the different combinations of input parameter values for these two input parameters.

5.7.3 Influential Parameters Effect Experiment

This experiment shows the effects that the chosen number of robots and the layout configuration have on the KPIs. The difference is that the fixed settings parameter experiment fixates the number of robots to 8, whereas this extra experiment varies the number of robots between 1 and 25 and the layout configurations. This experiment tries to show the minimum required number of robots per layout configuration, to comply with all the optimal KPI when the other input parameters such as battery charging rate and the battery level threshold are kept constant and considered to represent 'normal conditions'. For the 1624 bags scenario, 100 unique experiments are performed, however due to time and resource constraints these experiments are only replicated once to give a rough idea on the relation between these input parameters and the KPIs. Each configuration is combined with one to 25 robots and tested on the KPIs. For the 4290 bags scenario, 201 unique combinations are tested, also replicated once. Each combination is tested with 1 to 35 robots. Appendix K.5 shows the descriptive statistics for the 1624 scenario for the influential parameters effect experiment and Appendix K.6 shows the descriptive statistics for the 4290 bags scenario.

Table 15 shows for each layout configuration the minimum number of bags needed to comply with the desired thresholds of both the number of bags to be handled as the average process time of bags and the percentage of mishandled bags. It has become apparent that the KPI percentage of mishandled bags is the most important KPI for determining the number of robots needed. For the 1624 bags scenario for examples, 11 bags are needed in the central areas configuration in order to comply with the threshold for the percentage of mishandled bags. For the 4290 bags scenario, the experiment was extended to 96 robots instead of 35, as the value for the percentage of mishandled bags did not drop below the 0.02% in the one to 35 robots range. When testing the 36-96 robot range and thus adding 61 experiments, the 0.02% was never reached. In fact, the lowest percentage reaches was 8.72% with 47 robots.

Table 15 -Number of robots required to comply with KPI thresholds

	KPIs					
	# Handled Bags		**Average process time of bags**		**% of mishandled bags**	
Configuration	**1624 bags**	**4290 bags**	**1624 bags**	**4290 bags**	**1624 bags**	**4290 bags**
Central areas	6 robots	17 robots	9 robots	25* robots	11 robots	>96 robots
Four horizontal areas	6 robots	15 robots	8 robots	18 robots	10 robots	20 robots
Six horizontal areas	5 robots	14 robots	7 robots	16 robots	9 robots	18 robots
Six vertical areas	6 robots	15 robots	8 robots	18 robots	10 robots	21 robots

* depending on the seed, as just one replication is done for the experiment, no solid conclusions can be drawn for this output metric.

Figure 30 shows four histograms for the 1624 bags scenario with the number of robots on the x-axis and separate bars in the histogram for each layout configuration.

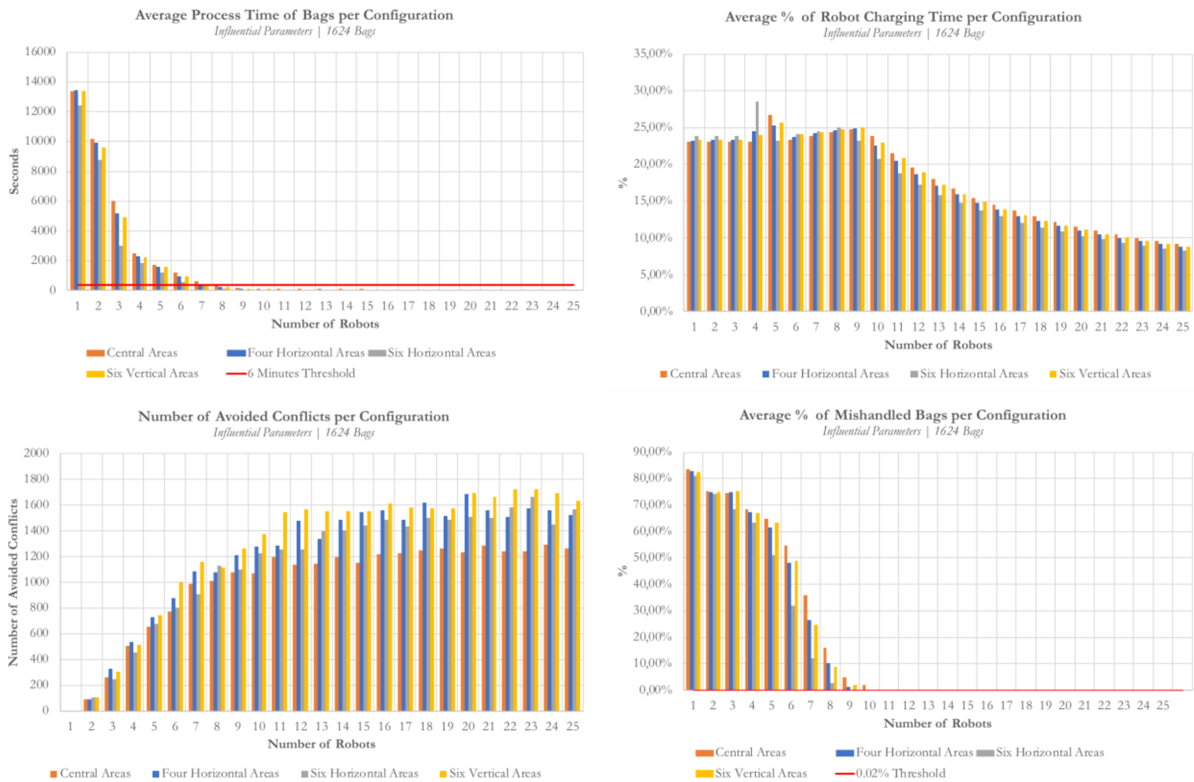

Figure 30 - Histograms of Influential Parameter Experiment with 1624 bags. Results for 1 replication

The top left histogram in the figure shows the average process time of bags per configuration in the 1624 bags scenario. From this histogram, it becomes clear that the more robots are used, the lower the average process time of bags becomes. However, after a certain point, the addition of an extra robot does not contribute a great deal to the reduction in average process time of bags. For the 1624 bags scenario, having 9 robots causes the average process time to be below the set threshold for all configurations. Adding an extra robot does decrease the average process time but cost estimations for the purchase of an additional robot have to show if it is worth it.

The top right histogram shows the average percentage of robot charging time per configuration. This histograms shows a downwards trend starting from approximately 10 robots in each configuration. This trend can seem counterintuitive, but it shows that the more robots are present, the less time is needed to charge. This is probably caused by having a lot of robots standing idle with a full battery. As these robots do not have to charge as it could be that they haven't been used, they influence the value of the counter as they are included in the calculation of the average charging time as well.

The bottom left histogram shows the number of avoided conflicts per configuration. From this upwards trend it can be concluded that robots get in each other's way more when the number of robot increases. In this histogram however, there is a clear difference visible between the configurations. The central areas configuration shows a significantly lower number of potential conflicts than the other configurations, regardless of the number of robots present. This can be explained by the manoeuvring space that this layout provides. As the charging and storage positions are located at the top and bottom of the layout as seen from above, robots have a larger open space to move around. The other areas have the same dimensions but are located in such a way that the robots have less manoeuvring space near the incoming conveyor belts, resulting in different shortest path possibilities.

The bottom right histogram shows the average percentage of mishandled bags per configuration. A clear downward trend is visible here and from 11 robots onwards, the value for this KPI drops below the 0.02% threshold. Adding an additional robot does not decrease this percentage even more, as it reaches a value of 0.00% in all configurations with 11 robots already.

When it comes to the layout configurations, the six horizontal areas configuration has the best value for the number of handled bags in runtime, the average process time of bags and the average percentage of mishandled bags. From this it can be concluded that this configuration provides the robots to get to the bags at the incoming conveyor belts fastest, resulting in a low value for the named variables. The central areas configuration is the most beneficial when the average percentage of mishandled bags is considered the most important KPI. A relatively high number of potential conflicts occur in the six horizontal areas layout, as the manoeuvre space for the robots is small near the incoming conveyor belts. The ratio loaded trips and empty trips is also the best for the six horizontal areas configuration. It is assumed that this is because the charging and storage positions are relatively the closest to the incoming conveyor belts, resulting in a minimal detour for robots that want to go to one of these positions.

The same histograms are also made for the 4290 bags scenario and are visible in Figure 31. The same logic as for the 1624 bags scenario applies here. The only major difference is that when the central area configuration is combined with one to 35 robots, an acceptable mishandled bags percentage is not reached. This specific configuration is further investigated for one to 96 robots and the mishandled bags percentage did never drop below the 0.02% threshold. This leads to the conclusion that for the 4290 bags scenario, the central areas configuration is not able to perform sufficient under this combination of input parameter values.

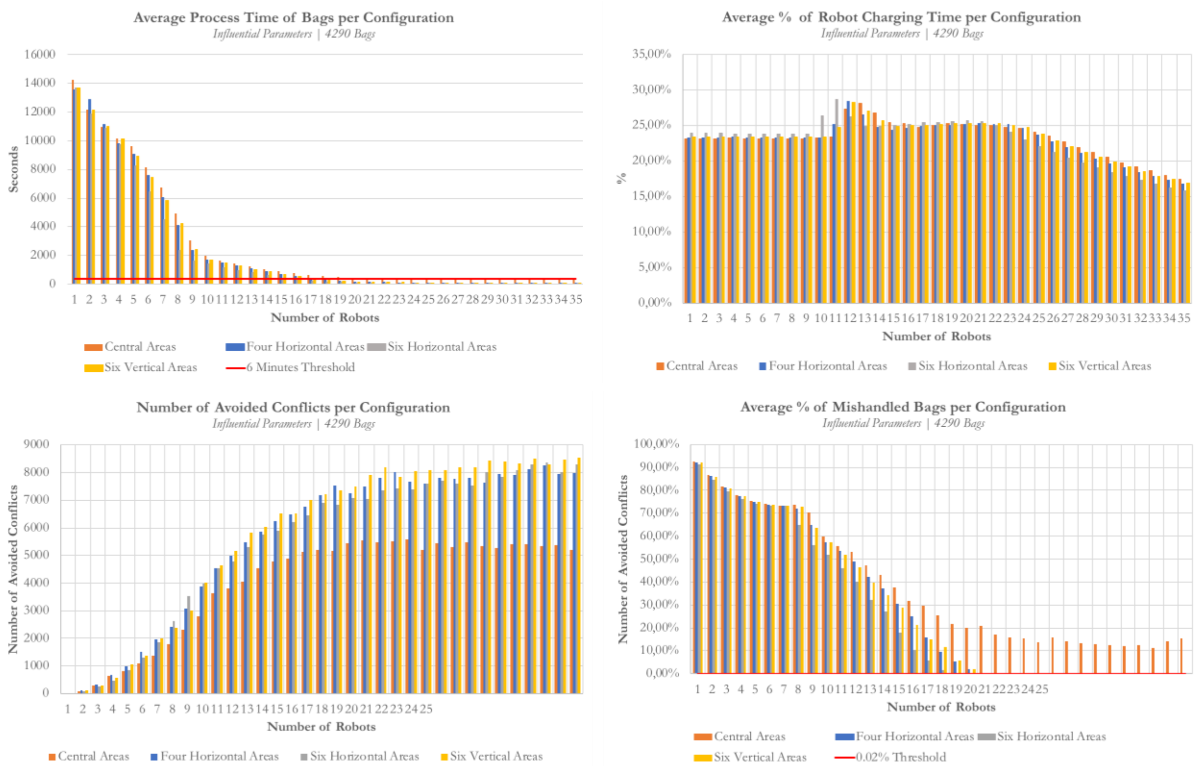

Figure 31 - Histograms of Influential Parameter Experiment with 4290 bags. Results for 1 replication

5.9. Conclusions on the Experiment Results

Three types of experiments have been executed; a fixed settings experiment, a parameter sweep experiment for both the 1624 and the 4290 bags scenario and an experiment to further investigate two most influential input parameters, being the number of robots and the layout configuration.

The fixed settings experiment shows promising results in terms of the average process time of bags. However, the specific combination of input parameter values appeared to be infeasible as the number of mishandled bags was too high with these values. The standard deviation due to stochastic variations in this experiment is relatively low, which shows that the number of replications needed for running the model with the exact same parameter settings does not need to be very high. To find a feasible combination of input parameter values, and to determine the most important design parameters and their optimal values, a parameter sweep experiment is executed.

The parameter sweep experiment showed that the baggage robot concept can be used in the sorting process of a baggage handling system at a medium-sized regional airport operating in a point-to-point network. However, under extreme circumstances – simulated by entering 4290 bags into the system – the baggage robot concept is not able to perform when a central areas layout configuration is chosen. Which exact parameter settings independently cause specific KPI values is hard to say as only a limited number of input parameter values are tested, based on a sample created by using Latin Hypercube Sampling. Data and experience with the model showed the two most important design choices in this baggage robot concept: number of robots and layout configuration. To investigate the effects of these two input parameters, a third experiment is executed, combining these two KPIs.

The influential parameter experiment links the two key design choices with the implications on the performance of the baggage robot concept in the sorting process of a baggage handling system. When it comes to the layout configuration, the main conclusion is that the closer the charging and storage positions are to the incoming conveyor belts, the less time bags have to wait at the end of the conveyor belt to be picked up and he lower the average process time of bags. Also, the percentage of empty trips is reduced as the detour for robots to go to one of the charging and storage positions is almost none. However, having the charging and storage positions located close to the incoming conveyor belts, reduces the manoeuvre space of the robots, resulting in more conflicts to be avoided than is the case in for example the central areas configuration.

When it comes to the regularity hypothesis discussed in section 5.1, it can be stated the hypothesis is accepted as regularity is found in the experiment results. The average process time of bags decreases as the number of robots increases, the number of avoided conflicts increases when the number of robots also increases, and the average percentage of mishandled bags decreases when the number of robots increases. This regularity is found and has appeared to be stable under different scenarios.

6. Conclusion and Discussion

The aim of this thesis has been to provide a conceptual design of future baggage handling systems based on the use of autonomous transport robots.. This concept level design has been tested and evaluated in an agent-based model. This chapter firstly addresses the research questions to draw the conclusions of this research. Secondly, all aspects of the research are discussed in a structured manner. This chapter ends with ideas and recommendations for future research on the topic of simulating and evaluating the baggage robot concept.

6.1. Conclusion: Answers to the Research Questions

This section answers the research questions as defined in section 1.2.2. Firstly, the five sub-research questions are answered. This section concludes with the answer on the main research question.

(1) What is the current state of baggage handling systems and transport robots?

This research question is answered in Chapter 2. When it comes to conventional baggage handling systems, dropped baggage undergoes five processes: (i.) security screening, (ii.) sorting, (iii.) makeup, (iv.) transporting to an aircraft and (v.) loading into an aircraft.

The dropped passenger bag is placed on a conveyor belt. From this conveyor belt the bag flows through a complicated system of connected conveyor belts until it reaches a makeup station. At the makeup station, the bag is manually put into in a baggage cart that is driven to the aircraft. Different elements of the conventional baggage handling system are currently improved. This can range from automated sorters to integrated screening solutions. But not only are elements added or automated in the conventional baggage handling system, the physical conveyor belts are also optimised, to make them more energy efficient for example.

The most recent innovation in the field of baggage handling systems is Vanderlande's FLEET system that combines conventional baggage handling systems with automated guided vehicles. Automated guided vehicles are a very common, widely used transport system. This driverless transport system is used for moving materials horizontally from one location to another. Automated guided vehicles are widely used in several industries, like warehousing and container handling. Since the introduction of FLEET, automated guided vehicles are finding their way to baggage handling systems in airports. In FLEET the complex system of connected conveyor belts is replaced by automated guided vehicles – transport robots – that carry individual baggage items all the way, from the drop-off facilities to the makeup stations.

(2) What Key Performance Indicators (KPI's) are relevant in assessing the capacity and continuity of a baggage handling system that makes use of autonomous individual transport robots?

This research question is answered in Chapter 3, section 3.4.4. The KPIs described in this section are specific for the sorting process of baggage handling systems that use individual transport vehicles, as it is argued that transport robots are expected to be the most valuable in this process. Not all mentioned KPIs are therefore applicable to or relevant for conventional baggage handling systems using conveyor belts. The rationale for the target values of the KPIs listed below is described in section 3.4.4.

The key performance indicators that have been identified are the following:

- *Average process time of bags*: the average time that elapses between the moments of the bag being offered for transport at the entrance of the sorting area and the bag arriving at the exit of the sorting area. Based on existing practice at a medium-sized regional airport in the Netherlands, the value of this KPI should not exceed six minutes, or 360 seconds.
- *Percentage of bags exceeding a norm time*: exceeding the norm time means that a bag arrives at a correct makeup station, but too late. Bags that have arrived too late cannot be loaded on

the aircraft before departure, resulting in bags being marked as mishandled. The percentage of mishandled bags should be as low as possible for a baggage handling system to be accepted for operation by ground handling parties and airports. Based on a case study at a medium-sized regional airport in the Netherlands, a value of 0.02% for this KPI is considered acceptable.

- *Average number of robots*: the number of robots present in the system to perform the transport and sorting task of the sorting process. This number should be no more and no less than the number of robots required for handling checked baggage at peak demand. The number of robots should be sparse, so as low as possible to comply with the optimal target values for the other KPIs.
- *Occupancy rate*
  - *Percentage of operational time while being loaded:* robots can be operational and non-operational. This KPI shows the percentage of the total time that a robot is executing a transport task. A low value of this KPI can indicate several things, including having too many robots in the system and inefficient charging strategies.
  - *Percentage of operational but empty trips by robots*: the percentage of empty trips should be minimal, indicating an efficient allocation of robots to bags. A value that is close to 50% is considered optimal for this KPI, as robots are only loaded when they are transporting bags from the entrance to the exit of the sorting process and need some time to go to charging or storage areas every now and then.
- *Percentage of charging time*: this KPI reflects charging efficiency. The amount of time a robot spends charging should be as low as possible, under the condition that the system complies with all the requirements and constraints of the system.
- *Number of conflicts avoided*: conflicts are not allowed to occur in a system in which robots transport bags as conflicts can lead to deadlock which can in its turn lead to system downtime. A low value for this KPI indicates that the chosen configuration of the system causes not too many imminent conflicts. However, a low value is not as strict requirement in itself, as long as all potential conflicts are avoided and conflict avoidance does not cause so much delay that the percentage of mishandled bags will increase.

(3) How can the autonomous individual transport robot concept be used in baggage handling systems?
This research question is answered in Chapter 3 'Baggage Robot Concept' by combining the existing elements of baggage handling systems and transport robots and improving them. Conventional baggage handling systems at airports use a complicated network of conveyor belts to transport bags from drop-off to the exit of the baggage handling area, from where bags are transported to the aircraft. The disadvantages of using a system of conveyor belts (which include being difficult to expand and difficult to relocate) may be overcome by the flexibility that is offered by autonomous individual transport robots. The disadvantages of conveyor belts appeared to be most present in the sorting process of the baggage handling chain. It is therefore concluded that the use of an alternative to a conveyor belt system – like individual transport robots – will be most beneficial in the sorting process. The baggage robot concept differs from the FLEET concept as it has more autonomy built into the robots. The robots are more intelligent and able to find their own paths in an open space, without being bound to an infrastructural grid. Given the current state of robot technology, it is questionable if the proposed type of autonomous robot is already available on the market. Yet, for future airport operations, autonomy provides more benefits, as it is easier to buy or insert one additional robot into the system. As long as it is an identical robot, it will be able to find its way through the area. In this way, the baggage robot concept is more flexible than an AGV system, as when an AGV system needs to be increased, new routes and predefined paths have to be established before the system can be operational again.

(4) How can the performance of a baggage handling system that makes use of autonomous individual transport robots be predicted and evaluated?

This research question is answered in Chapter 4. Using agent-based modelling a simulation model of the baggage robot concept in the sorting process of the baggage handling system can be run. The most important reason to use agent-based modelling is the autonomous nature of the individual transport robots considered in this research. Agent-based models are able to explicitly model the complexity that arises from individual actions and interactions that happen in such a system. The discrete entities – agents – present in such models are designed to mimic the behaviour of their real-world counterparts. Section 4.4.2. describes the requirements, constraints and assumptions used to develop an agent-based model of the sorting process of the baggage robot concept.

By means of a hybrid control approach, a simulation model is developed that is able to simulate the implementation of the baggage robot concept in the sorting process of baggage handling systems. In the simulation model, an algorithm has been developed that enables the transport robots to autonomously calculate and follow the shortest path while iteratively avoiding obstacles. By using multiple verification methods and expert validation, the model is verified and face validated and can be used to perform experiments. It is important to note that only the sorting process is modelled. This has been a scoping choice.

In addition, the use of individual transport robots is assumed to be the most beneficial in this specific process. The simulation model developed in this thesis therefore is not able to evaluate and predict the performance of the baggage robot system as a whole.

(5) What does the performance of a baggage handling system with autonomous individual transport robots look like?

This research question is answered in section 5.8 and 5.9. In this research, the main design choices for the baggage robot concept appear to be:

- the number of robots and
- the type of layout configuration

By executing three types of experiments the performance of the baggage robot concept in the sorting process is evaluated for two different scenarios. Scenario 1 has 1624 bags inserted into the system according to normally distributed arrival patterns. In Scenario 2 there are 4290 bags that are inserted following the same distribution. By executing these experiments, regularity was found for the KPIs. The average process time of bags decreases when the number of robots increases, for every configuration. The number of avoided conflicts increases when the number of robot increases and the average percentage of mishandled bags decreases when the number of bags increases. These regularities are found for all the configurations in both scenarios. The only layout that was unable to handle 4290 bags while having an acceptable percentage of mishandled bags is the 'central areas' configuration. The 'six horizontal areas' configuration appears to offer the best values for the KPIs considered.

Table 16 shows the number of robots required per layout configuration and per scenario for the three main KPIs. For example, to handle all 1624 bags in the 'central areas' layout configuration, at least 6 robots are necessary. To comply with the threshold for the average process time of bags in the 4290 scenario with a 'six vertical areas' layout, 18 robots are necessary. The same reasoning holds for the other values in the table.

Table 16 - Copy of Table 15 - Number of robots required to comply with KPI thresholds

	# Handled Bags		Average process time of bags		% of mishandled bags	
Configuration	**1624 bags**	**4290 bags**	**1624 bags**	**4290 bags**	**1624 bags**	**4290 bags**
Central areas	6	17	9	25*	11	>96
Four horizontal areas	6	15	8	18	10	20
Six horizontal areas	5	14	7	16	9	18
Six vertical areas	6	15	8	18	10	21

* depending on the seed, as just one replication is done for the experiment, no solid conclusions can be drawn for this output metric.

Considering the layout configuration, the main conclusion is that the closer the charging and storage positions are to the incoming conveyor belts, the less time bags have to wait at the end of the conveyor belt to be picked up and the lower the average process time of bags is. The percentage of empty trips is also reduced as the detour for robots to go to one of the charging and storage positions is almost none. However, having the charging and storage positions located close to the incoming conveyor belts, reduces the manoeuvre space of the robots, resulting in more conflicts to be avoided than is the case in for example the central areas configuration. This might well increase the risk of a collision in a practical baggage handling system.

By combining the answers on the sub-research question and knowledge gained while working on this research, the main research question can be answered:

In what way is it feasible to dynamically alter the floor plan and desired capacity of airport baggage handling systems by making use of autonomous individual transport robots?

By implementing the baggage robot concept, a step is made in dynamically altering the floor plan and desired capacity of airport baggage handling systems. The performance of this baggage robot concept has been reflected upon using the simulation results. Exact design parameters or values for these parameters cannot yet be given due to the limitations of the developed simulation model and the lack of reliable data available for a far-future concept. Yet, the number of robots and the layout configuration were found to be the most important design parameters. The number of robots should be such that the average process time of bags does not significantly decrease when introducing an additional robot to the system. The layout should be such that the route the robots travel is as short as possible, while at the same time ensuring the robots have enough manoeuvre space to reduce the number of possible conflicts. An example of such a layout that was chosen in this research is the 'six horizontal areas' layout, in which the charging and storage location of robots is located close to the incoming conveyor belts. Depending on the number of bags to be handled and the arrival pattern of these bags, a different minimum number of robots is necessary. When taking all the KPIs into account, this number should be higher (because of the determining KPI 'percentage of mishandled bags') than in case the other KPIs do not play a role and the acquisition costs of robots are the main determinant.

This thesis has been a first exploration in integrating autonomous robot systems in baggage handling systems, contributing to future proof and cost efficient operations.

Contribution of the thesis

The contribution of this thesis is twofold. Firstly, the contribution to airport operations in practice has been to sketch the possibilities, benefits, and disadvantages in the next phase of development of future baggage handling systems. Currently, the most advanced baggage handling system is FLEET – not guided by fully autonomous reasoning of the robots, but by predefined paths. This thesis has shown the way forward in integrating truly autonomous technology in baggage handling systems, by proposing a high-level design and evaluating the performance of this baggage handling system.

The scientific contribution of this thesis is in showing how to construct and simulate low-detail, flexible agent-based models to evaluate the performance of baggage handling systems. Agent-based modelling has been the most suitable modelling methodology due to the autonomous, individual nature of the robots. This thesis has shown how to integrally design a baggage handling system grounded in practical experience and requirements to its performance and at the same time demonstrating how to evaluate its performance.

6.2. Reflection on the Research: A Discussion on. …

This report is structured to describe the performed research on different aspects. This section starts with a discussion on the baggage robot concept as introduced in Chapter 3. After discussing this concept as a whole, the modelling phase and the resulting simulation model - on the baggage robot based sorting process - is discussed. With the simulation model, several experiments are executed, resulting in data that have been analysed.

6.2.1. The Baggage Robot Concept

In this paragraph the baggage robot concept is discussed. Chapter 3 has elaborated on the baggage handling concept as a supplement to or a replacement of conventional baggage handling systems at airports. The main aim of the baggage robot concept is to replace conveyor belts in the baggage handling system, to make it possible to dynamically alter the floor plan and the desired capacity of a baggage handling system. Vanderlande has recently introduced a system called FLEET that tries to accomplish the same. The difference between FLEET and the baggage robot concept as proposed in this research is the level of autonomy of the transport robots. In FLEET, the transport robots are automated guided vehicles, meaning they have to follow a predefined path. The baggage robot system however proposes individual transport robots that are more intelligent and capable of moving around the baggage handling area autonomously. The advantage of this approach is that individual robots can intelligently plan and re-plan their shortest routes, also when confronted with other robots crossing their paths at the same time. This concept therefore has an approach that is different from FLEET.

In this research, numerous assumptions have been made on the technical feasibility of these more intelligent and autonomous robots. For example, it is assumed that intelligent sensing capabilities can be integrated in the small robots and that the robots are capable of (almost) fully autonomous reasoning. However, this research did not include an exploration on the actual technical feasibility of these robots. Furthermore, the timescale and costs necessary to develop robots or a prototype of the robots that have these increased capabilities is not considered in this research. This means that it is not possible yet to estimate on which timescale the proposed baggage robot concept could be built into a prototype to further test the system.

As this research focuses on medium-sized regional airports, the description of the baggage robot concept in Chapter 3 pays no attention to the specific requirements of odd-sized and transfer baggage. This has been excluded from the research scope described in section 1.2.3. The consequence is statements on the use of the baggage robot concept in other types of airports cannot be made and that this research cannot provide specific statements on the baggage robot concept for airports in general.

Furthermore, the timescale and costs necessary to research and develop robots or a prototype of the robots that have these increased capabilities are not considered in this research. It looks as if it would be technically possible already to integrate autonomy into small-scale robots, considering recent developments on self-driving cars and e.g. Boston Dynamics robots performing a wide range of incredibly difficult tasks. It would thus be more of an economic investment-related challenge to build truly autonomous baggage robots than it would be a technological innovation challenge. This dependence on investment in baggage robot technology means that it is not possible yet to estimate with any precision on which timescale the proposed baggage robot concept could be built into a prototype to further test the system.

Furthermore, the timescale and costs necessary to research and develop robots or a prototype of the robots that have these increased capabilities are not considered in this research. It looks as if it would be technically possible already to integrate autonomy into small-scale robots, considering recent developments on self-driving cars and e.g. Boston Dynamics robots performing a wide range of incredibly difficult tasks. It would thus be more of an economic investment-related challenge to build truly autonomous baggage robots than it would be a technological innovation challenge. This dependence on investment in baggage robot technology means that it is not possible yet to estimate with any precision on which timescale the proposed baggage robot concept could be built into a prototype to further test the system.

6.2.2. Modelling and the Resulting Simulation Model

This paragraph reflects on the chosen modelling methodology – agent-based modelling – and discusses the benefits and disadvantages of the simulation model used to evaluate the sorting process of the baggage robot concept. Appendix L elaborates on how first-time users can use the developed simulation model and makes suggestions on how the model can be altered to serve different purposes.

Modelling

The process of designing and implementing the model has helped enormously in making the baggage robot concept more explicit and understanding all the aspects that play a role in designing such a concept. Modelling has made it possible to make a precise conceptual design of the baggage robot concept. However, this research only focuses on the sorting process of a baggage handling system at a medium-sized regional airport operating in a point-to-point network. The two other major processes (security screening and the makeup process) in the baggage handling system are not included, meaning the model is not able to show the performance of the baggage robot concept in the entire baggage handling process.

To translate the developed conceptual model into a working simulation model, agent-based modelling is used. The implementation of the agent-based model was done in NetLogo. Reflecting on these modelling decisions, it was the right choice to use agent-based modelling and NetLogo, as the model has naturally allowed for modelling autonomous robots. The NetLogo software is clear on which information is available in whose (autonomous) mind. Furthermore, NetLogo is well suited for building relatively small and flexible models and is open source software, making it relatively easy for others to use the developed model. A disadvantage of NetLogo is that it is not fast, especially compared to agent-based models that are for example directly programmed in Java. This restricts the researcher in running many variations and replications, reducing the statistical significance of the model results. When more detailed designs of the proposed baggage robot concept have been completed, it may be worthwhile to develop a more detailed agent-based model, making the generated results more reliable.

The model has been face validated by three experts. As discussed in section 4.7.2, the use of expert validation does not imply that the model can be considered fully validated. To strengthen the validity of the developed model, a discrete event model or an agent-based model of the current baggage handling system with conveyor belts can be built. This additional model can then be used for comparison with the model developed in this research, to pinpoint the most significant differences. However, this still does not fully validate the developed model, as true validation checks if the model is a good representation of the real world system. In this case, the real world system does not exist yet. Reflecting on the nature of current BHS systems, it would likely be more natural to use a discrete event model, although it is more difficult to compare two models built in two different modelling methods.

Simulation Model

The developed simulation model of the baggage robot concept is an abstraction of the real world concept. This means that assumptions and simplifications have to be made. In the simulation model, only four incoming conveyor belts and four makeup stations are programmed. The reason to select four makeup stations is because the data used to construct an arrival pattern required only four makeup stations when the flights were manually scheduled. The four incoming conveyor belts were arbitrarily chosen as they don't have to represent the number of bag-drop facilities since the model skips the security screening process. The four incoming conveyor belts therefore represent four points where the security process can insert security-cleared bags into the sorting process. A drawback of the simulation model in its current state is the inability to easily alter the number of incoming conveyor belts and makeup stations to experiment with, as well as the locations of these facilities in the baggage handling area. Such additions can help in understanding and showing the impact of a different number and location of these facilities on the performance of the baggage robot concept in the sorting process.

As mentioned, the simulation model only represents the sorting process in the baggage handling system of an airport that operates in a point-to-point network. To fully test the performance of the baggage robot as an addition to or replacement of a conventional baggage handling system at such an airport, the security process and makeup process need to be included as well. In all of these processes, odd-sized baggage needs to be accommodated for as well. To simulate the baggage robot concept in a different airport setting, aside from the mentioned additions, transfer facilities need to be added as well. When additional KPIs such as the waiting time for passengers that want to drop off their bag at one of the bag-drop facilities are desired, the assumption that the buffer capacity of the incoming conveyor belts is infinite needs to be changed to a limited capacity. A limited buffer capacity should be considered to investigate the waiting time of bags on the belt, which has an effect on the bag offering process. By restricting the buffer capacity, research can be done on the development of passenger queues at bag-drop facilities.

Another important assumption that is made in the simulation model is that the arrival pattern is considered to follow a normal distribution with a mean of 40 minutes and a standard deviation of 30 minutes. In this research, no variations to this arrival pattern are considered. The results generated by this simulation model can therefore not be generalized to all possible arrival patterns. To investigate the impact of the arrival pattern on the performance of the baggage robot concept in the sorting process, it is recommended to experiment with the gamma distribution and different values for the mean and standard deviation. To make the arrival pattern as realistic as possible, data need to be obtained from different airports of different sizes worldwide. By doing so, the accuracy of the KPI results can be increased. Currently, the accuracy of the KPI results is not too high because a normal distribution with a specific mean and standard deviation is assumed and the number of bags that are entered into the system on a daily basis is only varied twice (1624 bags and 4290 bags). When investigating the implementation of the baggage robot concept at a different type and size of airport, these values have to be adjusted.

Important to mention is that in this research, a mishandled bags percentage of 0.02% is assumed. This means that if during the sorting process 0.02% of the bags are marked as 'mishandled', there is no leeway for errors in the security and makeup process. The feasible KPI scores are therefore also influenced by the scoping choice of only considering the sorting process.

When it comes to the software implementation, simplifications and assumptions have also been made. In the developed simulation model, robots are only positioned in the middle of a patch. This implies the use of a buffer zone between robots because a robot does not occupy a whole patch. However, this buffer zone size might be variable, to investigate the impact on for example process time. When moving around, the robots have to construct and follow a shortest path. This research used the A* algorithm to calculate the shortest path for each robot to each destination.

In the software implementation, conflict and deadlock resolution were relatively difficult to implement. In the current model implementation only one way to deal with this has been implemented. There are however other ways of implementing these autonomous procedures. The current implementation uses priority rules for transport robots carrying a bag and decides randomly which empty transport robot can go first after the loaded robot has passed in case of imminent collisions. The other involved robots have to wait until they can proceed to follow their shortest path. This implies that a part of the robots is waiting – even though it is only for just a time step of one second. This could have implications for the simulation results. The baggage robot concept might thus – accepting the relevant assumptions and simplifications – even be slightly more effective than becomes apparent from the current experiment results, as the waiting time caused by imminent collisions can be less than one second.

6.2.3. Simulating and Data Analysis

This paragraph discusses the possibilities for simulation that the current model allows for. Next to this, the merits of the current data analysis are discussed.

Simulating

While simulating the baggage robot concept in the sorting process, each tick is assumed to represent one second. However, one tick per second makes the simulation too laborious. In the current low detail implementation this was a way to take into account that robots do not physically bump into each other, and at the same time to make sure that all robots move at the right speed. While running the model, the shortest path calculations for every robot slowed down the simulation time significantly. Although runtime calculations have not been performed, this is the only computation done in the model that is non-trivial in terms of runtime.

The simulation speed decreased even more when the number of robots or the number of bags inserted into the system was larger. This can be explained by the fact that many more optimization calculations are needed in such cases to run the model. As the model was relatively slow, it has not been possible to perform full sweeping simulation experiments to investigate the full output space in far more detail. This does have implications for the data analysis, but given the time and resources constraints in this research, Latin Hypercube Sampling is considered a good solution to overcome these limitations.

Data Analysis

The developed simulation model has been used to perform four types of experiments amounting to 1252 replications. This is a relatively small number for a stochastically varying agent based model. It would have been advantageous to perform far more experiments if resources and time available would have allowed for this.

As it has been quite difficult to perform a balanced set of experiments, Latin Hypercube Sampling was used. The data analysis therefore is performed on three experiment designs, resulting in 1 experiment for the first type which is replicated 50 times, 30 experiments for the second type which is replicated 30 times and 301 experiments for the third type (4 times 25 unique experiments for the 1624 scenario, 3 times 35 unique experiments for the 4290 scenarios plus an additional 96 unique experiments for the 4290 scenario combined with the central areas configuration). The experiments of the third experiment design are only replicated once due to speed, time and resource limitations. Given these limitations, it should be argued that the level of detail of the results corresponds to the level of detail made possible by the nature of the experiments. From the data analysis, mainly qualitative conclusions have been found about the inner workings of the proposed baggage robot concept in the sorting process, the order of magnitude of the KPI scores and resulting prospective feasibility of the baggage robot concept.

Validity of the model and the simulation
The validity of the current simulation model has been tested by performing an expert-based review of the model. This is a method that is limited in fully assessing the validity of both the models and the results. To increase the validity of the model, it had to be compared to a similar model developed by a different research group, by comparing it to a similar model on the conventional baggage handling system or to a similar model in which the robots follow predefined paths such like AGVs. When more detailed data on the performance of current conventional baggage handling systems and for example FLEET can be obtained, conclusions on the results can be substantiated better. However, it must be argued that the level of validity of the current model does fit the design stage that the baggage robot concept currently is in.

6.2.4. Towards the Success of the Baggage Robot Concept
Concluding the discussion it can be stated that from the current simulation results it has become apparent that the designed baggage robot concept system is feasible in dynamically altering the floor layout and desired capacity of the sorting process of baggage handling systems. The exact design parameters – such as the number of robots and the chosen layout – should be made more precise in later iterations of the conceptual baggage robot concept design, and subsequent iterations of the model-based evaluations of these concepts.

This research gives insight into what rough design choices have to be made when researching and developing a baggage robot concept. The nature of implications of these design choices becomes apparent from qualitative guidance on how to continue the design process in the section on future research. The low-level of detail simulation that has been used in this research might very well be an interesting instrument to evaluate other autonomous robot-based logistic systems in different industries too.

6.3. Future Research
This section concludes with proposed topics for future research on the baggage robot concept as a supplement to or a replacement of conventional baggage handling systems. The section is structured in the same way as section 6.2. It reflects on which design aspects have not been taken into account in the high level concept design of the baggage robot concept, and on the current level of knowledge that is available on the strengths and weaknesses of the proposed concept.

6.3.1. The Baggage Robot Concept
In the part of the discussion on the baggage robot concept itself, it was stated that numerous assumptions have been made on the technical feasibility of the robots. In this research the robots are considered to be more intelligent and autonomous than the robots used in Vanderlande's FLEET concept. A possible follow-up research should therefore lean more towards autonomy than FLEET does at the moment.

A suggestion for future research is to explore the technical feasibility of the proposed robot types. One important starting point can be to research the sensing capabilities of the transport robots. To do so, inspiration can be taken and lessons can be learned from innovations in the field of self-driving cars. Although the self-driving car technology is still being developed, it can help to improve the individual transport robot systems. The state-of-the-art when it comes to self-driving cars is that they perceive the world by combining information coming from different types of sensors, including cameras, radar and LIDAR. LIDAR is a technique that is similar to radars and uses invisible light pulses to map its surrounding area. By combining the different types of sensors, the safety and reliability of autonomous vehicles can be increased as the sensors complement each other. The cameras are cheap and are able to detect markings on the infrastructure, but are unable to measure distances. This is where radars complement cameras, as they are able to measure distance and velocity. However, radars are not able to provide the vehicle with fine details on its environment. That is where LIDAR is useful, as it is able to do so. By combining the data from these three sensors, the vehicle is able to identify its surroundings and everything that's in it, including road markings, other vehicles, walls, and so forth. The robots in the baggage robot concept do not have to be as intelligent as self-driving cars, as they will be used inside and in a highly controlled environment. This means that sensors do not have to be able to for example detect snow or a plastic bag blowing across the area ("Special Report Autonomous Vehicles," 2018).

In this research, the individual robots play a central role. In their movements, they detect and react to other robots in the area, but do not share information continuously. The developments in self-driving or autonomous cars can help in shifting the individual optimization perspective – possibly leading to sub-optimal solutions and routes – to a perspective in which the operation of the system as a whole is optimized. This can for example be done by 'fleet learning', a process in which vehicles learn from each other's experiences by comparing the data they gathered while driving through the same area ("Special Report Autonomous Vehicles," 2018).

Increasing the capabilities and intelligence of vehicles – being autonomous cars or autonomous transport robots - will not necessarily eliminate the need for human assistance. Autonomous vehicles are programmed to obey defined rules but still can get stuck when unforeseen circumstances occur. Unlike humans, programmed vehicles are unable to bend the rules in such circumstances to for example drive around a blockage. Human controllers can give temporary permission to the vehicle to deviate from the set rules to resolve unforeseen and undesired situations ("Special Report Autonomous Vehicles," 2018).

As mentioned in the part of the discussion on the baggage robot concept itself, the timescale and costs necessary to research and develop robots or a prototype of the robots that have these increased capabilities and intelligence are not considered in this research. Future research on timescale and cost estimates for research and development can help in estimating on which timescale the proposed baggage robot concept could be built into a prototype system to further test it. It is recommended to test these prototypes on different types and sizes of airports to verify the suitability of the baggage robot concept in different circumstances. For smaller airports for example, it might be too expensive to buy highly modern autonomous robots, while for large hub airports it might be an interesting baggage handling system considering the expected lower operational costs over time.

Furthermore, the cost dimension of the proposed baggage robot concept has been omitted completely. Both capital expenditure and operational expenditure costs are important to take into account in the development and operation of a baggage handling system. Such a cost analysis would first be highly instructive as to compare the current conventional baggage handling system cost with the prospective cost of the proposed autonomous system. The costs of conventional systems include costs over time, also when confronted with increasing demand and thus bulk investment cost. The costs of the proposed autonomous system includes cost over time, also when confronted with increasing demand thus incremental investment cost of buying extra robots.

6.3.2. Modelling and the Resulting Simulation Model

Modelling

During the modelling process, scoping choices have been made resulting in a heavy NetLogo model of the first high-level design of the sorting process of the baggage robot concept. For future research, it is recommended to develop a more detailed agent-based model when more detailed designs of the proposed (full) baggage robot concept have been completed.

To give a full picture on the performance of the baggage robot concept in the broadest sense, it would be interesting to extend the scope of the model to the security screening and the makeup process, the other two major processes in baggage handling. An addition that adds value but complicates the modelling process is to add holding positions for extra decision time in the screening process and storage for early bags.

Another additional model that can be developed to fully investigate the pros and cons of the baggage robot concept is a variation of the already developed model. A difference that can be made to see the impact on for example capacity is to include predefined paths in the model that the robots can follow. This additional model would then be an AGV-variant of the model of the baggage robot concept, developed with agent-based modelling.

When additional models on the conventional baggage handling system and a baggage system that uses AGVs would be developed, the performance of the baggage robot system can be compared and judged. In that case, it is recommended to also include the CAPEX and OPEX cost as well as energy consumption. In order to convince an airport to change its current way of handling baggage, these elements must be taken into account.

Next to that, it has been difficult to operationalise the overall performance of a given parameterised autonomous BHS system layout. The first component of such an overall analysis is if a given baggage handling system fulfils all feasibility conditions, this analysis has already been performed. However, it is not a priori clear if lower process time or lower numbers of mishandled bags should be aimed for. The relative importance of different KPIs is not elucidated yet. A proposed solution would be to integrate the KPIs into a multi-criteria analysis to compare this overall performance, wherein the weights of different KPIs would need to be calibrated by further desk study, expert knowledge or end-user experience.

Within the model, the collision and deadlock avoidance strategies can also be improved in future research. A suggestion to improve the collision and deadlock avoidance strategies is to take the modified voltage potential algorithm, known from free flight research, as an inspiration. In a situation in which more than two robots are involved in an imminent conflict, constraints imposed in the simulation model, like the limited space available because of walls, the number of robots, the fixed speed of the robots, the fact that a lot of robots simultaneously have to go to one specific point representing a makeup station, cause difficult to solve collision and conflict avoidance situations. Literature on free flight shows that imposing such constraints make solving the multiple robot problem near to impossible. A solution can be the modified voltage potential algorithm, in which entities – like aircraft or in this case robots – share information. An example of this is that one robot asks another robot to move in such a way that it can move itself and get out of a deadlock situation itself. By sharing information on e.g. their current situation and planned path, a deadlock situation can be avoided or resolved. It could be that new deadlocks exist, but by collaborating, these deadlocks can be resolved as well.

Simulation Model

When it comes to the simulation model itself, future research can add value to the existing model by implementing and experimenting with more incoming conveyor belts and makeup stations, as well as the locations of these facilities in the baggage handling area. This can help in researching the impact of a different number and location of these facilities on the performance of the baggage robot concept in the sorting process.

The construction of the shortest path in the simulation model can be optimized in future research by for example allowing for diagonal movements or free-moving, as the real world is not divided into patches. Appendix C.1. explains the difference diagonal movements can make in terms of the number of steps necessary to arrive at a point compared to the stepwise approach that is used in this research. Furthermore, the A* algorithm is only one of many algorithms for path planning. To investigate the effect of the choice for this algorithm, variations to the model have to be made in which a different algorithm is implemented. Examples of other algorithms include the Dijkstra algorithm, Basic Theta*, Phi* and the Greedy Best First Search.

6.3.3. Simulating and Data Analysis

Simulating

For future research, it would be good to perform runtime calculations to check exactly what part of the simulation took what amount of time. By doing so, parts that slow down the simulation model can be identified and might be optimized. It could be that the complexity of the algorithm slows the simulation down, or that the algorithm itself does not performs optimally. Identifying the cause and possibly optimizing it can help in simulating a full day or multiple weeks, if desired. While running the model, the simulation speed decreased even more when the number of robots or the number of bags inserted into the system was larger. This can be explained by the fact that many more optimization calculations are needed in such cases to run the model. As the model was relatively slow, it has not been possible to perform full sweeping simulation experiments to investigate the full output space in far more detail. This does have implications for the data analysis, but given the time and resources constraints in this research, Latin Hypercube Sampling is considered a good solution to overcome these limitations.

Data Analysis

Provided that the model has been strengthened and made faster in the future, there are many possibilities to perform more detailed and deep analyses that can (i.) increase trust in the results, and that can (ii.) increase the depth of results as found from the model. Multiple regression analysis or comparable statistical techniques might be used to investigate the relations between the variety of input (design) parameters and the output (KPI) metrics. More advanced machine learning techniques, such as Patient Rule Induction Method, might even be used to get to know the mapping between input space and output space in far more detail.

Validity of the model and the simulation

If in future research the concept is brought further, more detailed simulation methods should be used to prove more detailed results and evaluative hypotheses about the workings of the detailed design. To thoroughly validating the model, several of the validation methods as described in section 4.7.2 can be used to not only give the agent-based simulation model face validity, but fully validate it. A more validated model can increase the reliability of its results, but it can also increase the commitment of stakeholders for the baggage robot concept when these stakeholders are also involved in the validation steps.

The concept design and simulation experiments performed in this thesis have contributed extensively to understanding the complex challenges inherent to integrating individual transport robots in real-life baggage handling systems. By furthering the application of flexible agent-based modelling to the field of airport operations valuable practical lessons can be learned to fulfil the ambitions in airports of the future.

Bibliography

Amberber, E. (2014). How Amazon's Kiva robots shorten order fulfillment time – 30 minutes instead of hours. Retrieved March 13, 2018, from https://yourstory.com/2014/12/amazon-kiva-robots/

Arai, Y. (1999). Collision avoidance in multi-robot systems based on multi-layered reinforcement learning. *Robotics and Autonomous Systems*, *29*(1), 21–32. https://doi.org/10.1016/S0921-8890(99)00035-4

ATAG. (2005). The economic & social benefits of air transport. *Air Transport Drives Economic and Social Progress Air*, 28.

Azadeh, K., de Koster, M. B. M., & Roy, D. (2017). Robotized Warehouse Systems: Developments and Research Opportunities. *SSRN Electronic Journal*, 1–55. https://doi.org/10.2139/ssrn.2977779

Bae, H. Y., Choe, R., Park, T., & Ryu, K. R. (2011). Comparison of operations of AGVs and ALVs in an automated container terminal. *Journal of Intelligent Manufacturing*, *22*(3), 413–426. https://doi.org/https://doi.org/10.1007/s10845-009-0299-1

Baker, P., & Halim, Z. (2007). An exploration of warehouse automation implementations: cost, service and flexibility issues. *Supply Chain Management: An International Journal*, *12*(2), 129–138. https://doi.org/10.1108/13598540710737316

Beumer Group. (n.d.). Baggage handling made different a Head for heights.

Borshchev, A., & Filippov, A. (2004). From System Dynamics to Agent Based Modeling. *Simulation*, *66*(11), 25–29.

Business Insider UK. (2017). Amazon now has 45,000 robots in its warehouses. Retrieved March 13, 2018, from http://uk.businessinsider.com/amazons-robot-army-has-grown-by-50-2017-1?international=true&r=UK&IR=T

Calleam Consulting Ltd. (2008). Denver International Airport Baggage Handling System: An illustration of ineffectual decision making. *Calleam Consulting*, (August 2005), 10.

Cavada, J. P., Cortés, C. E., & Rey, P. A. (2017). A simulation approach to modelling baggage handling systems at an international airport. *Simulation Modelling Practice and Theory*, *75*, 146–164. https://doi.org/10.1016/j.simpat.2017.01.006

Clénet, D. (2010). Optimising Energy Efficiency of Conveyors. *Schneider Electric White Paper*, (January).

Cruse, T. A. (1997). *Reliability-Based Mechanical Design*. Boca Raton: CRC Press.

Dadyala, G. (2013). Short-Term Baggage Storage Systems Lead to Long-Term Success. *Consultant Perspective*, 6–7.

Dam, K. H. van, Nikolic, I., & Lukszo, Z. (2010). *Agent-Based Social Systems Volume 7 Editor in Chief : Series Editors :* (H. Deguchi, Ed.) (Vol. 9). Springer. https://doi.org/10.1007/978-94-007-4933-7

Delorme, P., Procter, J., Swaminathan, S., & Tillinghast, T. (1992). Simulation of a combination carrier air cargo hub. *Proceedings of Ihe 1992 Winter Simulation Conference*, 1325–1331.

Demag. (n.d.). AGV in Container Terminal. Retrieved from http://www.demagcranes-ag.de/files/content/sites/deutschland/files/Buehnenbilder/HaefenUndTerminals/AGV_1_schatten.jpg

Dubiel, B., & Tsimhoni, O. (2005). Integrating agent based modeling into a discrete event simulation. *Proceedings of the 2005 Winter Simulation Conference*, 1029–1037.

Egbelu, P. J. (1987). The use of non-simulation approaches in estimating vehicle requirements in an automated guided vehicle based transport system. *Material Flow*, *4*(1), 17–32.

Eilon, S., & Mathewson, S. (1973). A Simulation Study for the Design of an Air Terminal Building. *IEEE Transactions on Systems, Man, and Cybernetics*, *3*(4), 308–317. https://doi.org/10.1109/TSMC.1973.4309241

Enserink, B., Hermans, L., Kwakkel, J., Thissen, W., Koppenjan, J., & Bots, P. (2010). *Policy Analysis of Multi-Actor Systems* (1st ed.). The Hague: LEMMA.

European Commission. (n.d.). 2050AP Report Summary.

Eurostat. (n.d.). Thousand passengers handled in top airports monthly data 2015-2016. Retrieved October 11, 2017, from http://ec.europa.eu/eurostat/statistics-explained/index.php/File:Thousand_passengers_handled_in_top_airports_monthly_data_2015-2016.png

Evers, J. J. M., & Koppers, S. A. J. (1996). Automated Guided Vehicle Traffic Container Terminal Control. *Science*, *30*(I), 21–34.

Field, B. F., & Kasper, J. G. (1991). Automated Guided Vehicle.

Gaskins, R. J., & Tanchoco, J. M. a. (1987). Flow path design for automated guided vehicle systems. *International Journal of Production Research*, *25*(5), 667–676. https://doi.org/10.1080/00207548708919869

Goetz Jr., W. G., & Egbelu, P. J. (1990). Guide path design and location of load pick-up/drop-off points for an automated guided vehicle system. *International Journal of Production Research*, *28*(5), 927–941.

Gourgand, M., Xiao-Chao Sun, & Tchernev, N. (1995). Choice of the guide path layout for an AGV based material handling system. In *Proceedings 1995 INRIA/IEEE Symposium on Emerging Technologies and Factory Automation. ETFA'95* (Vol. 2, pp. 475–483). IEEE Comput. Soc. Press. https://doi.org/10.1109/ETFA.1995.496688

Graham, A., & Metz, D. (2017). Limits to air travel growth : The case of infrequent fl yers. *Journal of Air Transport Management*, *62*, 109–120. https://doi.org/10.1016/j.jairtraman.2017.03.011

Grenzebach. (n.d.). Automated Baggage Handling.

Grigora, R., & Hoede, C. (2007). Design of a Baggage Handling System. *Department of Applied Mathematics*, *1*(April 2007), 17.

Guizzi, G., Murino, T., & Romano, E. (2009). A Discrete Event Simulation to model Passenger Flow in the Airport Terminal. *Mathematical Methods and Applied Computing*, 427–434.

Haefner, L. E., & Bieschke, M. S. (1998). ITS Opportunities in Port Operations. *Transportation Conference Proceedings*, 131–134.

Hallenborg, K., & Demazeau, Y. (2006). Dynamical Control in Large-scale Material Handling Systems through Agent Technology. *IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2006. IAT '06*, 637–645. https://doi.org/10.1109/IAT.2006.59

Hallenborg, K., & Demazeau, Y. (2008). DECIDE: Applying multi-agent design and decision logic to a baggage handling system. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5049 LNAI, pp. 148–165). https://doi.org/10.1007/978-3-540-85029-8-11

IATA. (n.d.). Baggage Tracking. Retrieved November 1, 2017, from http://www.iata.org/whatwedo/ops-infra/baggage/Pages/baggage-tracking.aspx

IATA. (2016). IATA Forecasts Passenger Demand to Double Over 20 Years. Retrieved September 26, 2017, from http://www.iata.org/pressroom/pr/Pages/2016-10-18-02.aspx

Jäger, M., & Nebel, B. (2001). Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. *IEEE International Conference on Intelligent Robots and Systems*, *3*, 1213–1219. https://doi.org/10.1109/IROS.2001.977148

Janssen, M. A. (2005). Agent-based modelling. *Modelling in Ecological Economics*, 155–172.

Johnstone, M., Creighton, D., & Nahavandi, S. (2015). Simulation-based baggage handling system merge analysis. *Simulation Modelling Practice and Theory*, *53*, 45–59. https://doi.org/10.1016/j.simpat.2015.01.003

Johnstone, M., Le, V. T., Zhang, J., Gunn, B., Nahavandi, S., & Creighton, D. (2015). A dynamic time warped clustering technique for discrete event simulation-based system analysis. *Expert Systems with Applications*, *42*(21), 8078–8085. https://doi.org/10.1016/j.eswa.2015.06.040

Khamis, A., Hussein, A., & Elmogy, A. (2014). Cooperative Robots and Sensor Networks 2014, *554*, 31–52. https://doi.org/10.1007/978-3-642-55029-4

Klee, H. (1986). Some novel applications of a continuous system simulation language. *Computers and Industrial Engineering*, *11*(1–4), 385–389. https://doi.org/10.1016/0360-8352(86)90117-8

Kook, J., Mook, J., Yoshimoto, K., & Hwan, K. (2002). Designing guide-path networks for automated guided vehicle system by using the Q -learning technique q, *44*, 1–17.

Le-Anh, T., & De Koster, M. B. M. (2006). A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, *171*(1), 1–23. https://doi.org/10.1016/j.ejor.2005.01.036

Le, V. T., Zhang, J., Johnstone, M., Nahavandi, S., & Creighton, D. (2012). A Generalised Data Analysis Approach for Baggage Handling Systems Simulation. *IEEE International Conference on Systems, Man, and Cybernetics*, 1681–1687.

Lui, R., Nanda, R., & Browne, J. J. (1972). International Passenger and Baggage Processing at John F. Kennedy International Airport. *IEEE Transactions on Systems, Man and Cybernetics*, *2*(2), 221–225. https://doi.org/10.1109/TSMC.1972.4309096

Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*, *4*(3), 151–162. https://doi.org/10.1057/jos.2010.3

Majety, S. V., & Wang, M. H. (1995). Terminal location and guide path design in terminal based AGV systems. *International Journal of Production Research*, *33*(7), 1925–1938.

https://doi.org/10.1080/00207549508904790

Maki, D., & Thompson, M. (2005). *Mathematical Modeling and Computer Simulation* (1st ed.). Cengage Learning.

Marion, G., Scotland, S., Lawson, D., & Marion, G. (2008). An Introduction to Mathematical Modelling List of Figures.

Mas, I., & Kitts, C. (2010). Centralized and decentralized multi-robot control methods using the cluster space control framework. *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, (August), 115–122. https://doi.org/10.1109/AIM.2010.5695768

Mauro, F. (2017). *Towards the design of an effective and robust multi-robot parcel sorting system*. Delft University of Technolgy.

Mehta, N. (2014). How the baggage handling system works. Retrieved from http://www.livemint.com/Specials/53KK7gaj40CSeoli891hkO/How-the-baggage-handling-system-works.html

Montealegre, R., & Keil, M. (2000). De-Escalating Information Technology Projects: Lessons from the Denver International Airport. *MIS Quarterly*, *24*(3), 417–447. https://doi.org/10.2307/3250968

Mota, M. M., & Boosten, G. (2011). Extended Definition of Capacity in Airport Systems. In *ISNGI*.

Müller, T. (1983). *Automated Guided Vehicles*. Bedford: IFS.

Northwestern University. (n.d.). What is NetLogo? Retrieved February 1, 2018, from https://ccl.northwestern.edu/netlogo/docs/whatis.html

Nsakanda, A. L., Turcotte, M., & Diaby, M. (2004). Air cargo operations evaluation and analysis through simulation. *Proceedings of the 2004 Winter Simulation Conference, 2004.*, *2*(2000). https://doi.org/10.1109/WSC.2004.1371531

Odoni, A. R. (1980). Airport engineering. *Transportation Research Part A: General*, *14*(1), 67–68. https://doi.org/10.1016/0191-2607(80)90098-9

Olsson, A., Sandberg, G., & Dahlblom, O. (2003). On Latin hypercube sampling for structural reliability analysis. *Structural Safety*, *25*(1), 47–68.

Ottjes, J. A., Veeke, H. P. M., Duinkerken, M. B., Rijsenbrij, J. C., & Lodewijks, G. (2007). Simulation of a multiterminal system for container handling. *Container Terminals and Cargo Systems: Design, Operations Management, and Logistics Control Issues*, *468*, 15–36. https://doi.org/10.1007/978-3-540-49550-5_2

Özgün, O., & Barlas, Y. (2009). Discrete vs. continuous simulation : When does it matter? *27th International Conference of The System Dynamics Society*, (6), 1–22.

Papatheodorou, A., Graham, A., & Forsyth, P. (2016). *Aviation and Tourism*. Routledge.

Parker, L. E. (2009). Springer handbook of robotics. *Choice Reviews Online*, *46*(6), 46-3272-46–3272. https://doi.org/10.5860/CHOICE.46-3272

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information*

Systems, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Rauch, R., & Kljajić, M. (2006). Discrete Event Passenger Flow Simulation Model for an Airport Terminal Capacity Analysis. *Organizacija*, *10*(december), 652–660.

Riley, L. A. (2013). Discrete-event simulation optimization: A review of past approaches and propositions for future direction. *Simulation Series*, *45*(11), 386–393.

Ritter, J. B. (1992). Defining International Electronic Commerce Electronic Commerce. *Northwestern Journal of International Law and Business*, *13*(1), 1–29.

Robinson, G. L. (1969). Simulation models for evaluation of airport baggage-handling systems. In *Proceedings of the 3rd Conference on Applications of Simulation* (pp. 226–235).

Rowley, J. (2000). *The principles of warehouse design, Guideline No. 4* (2nd ed.). Corby: The Institute of Logistics and Transport.

Roy, D., & De Koster, R. (2014). Modeling and Design of Container Terminal Operations. *ERIM Report Series Research in Management Erasmus Research Institute of Management.*, *2017*(2013), 1–57. https://doi.org/http://dx.doi.org/10.2139/ssrn.2458048

SAE. (2008). utomated shuttle carrier expands Kalmar's cargo handling options. Retrieved from http://articles.sae.org/1752/

Sargent, R. G. (2010). Proceedings of the 2010 Winter Simulation Conference B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds. *Simulation*, (2001), 135–150. https://doi.org/10.1109/WSC.2010.5679148

Savrasovs, M., Medvedev, A., & Sincova, E. (2009). Riga Airport Baggage Handling System Simulation. *Proceedings of 23rd European Conference on Modelling and Simulation (ECMS)*, 384–390. https://doi.org/10.7148/2009-0384-0390

Schiphol Airport. (n.d.). Odd-Sized Baggage. Retrieved November 1, 2017, from https://www.schiphol.nl/en/page/odd-size-baggage/

Schultz, M., & Fricke, H. (2011). 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*, 438–447.

Schulze, L., Behling, S., & Buhrs, S. (2008). Automated Guided Vehicle Systems: a driver for increased business performance. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, *II*, 19–21.

Siebers, P. O., Macal, C. M., Garnett, J., Buxton, D., & Pidd, M. (2010). Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, *4*(3), 204–210. https://doi.org/10.1057/jos.2010.14

SITA. (n.d.). SITA reports improvement in rate of bag delivery.

Smith, R. D. (1998). Simulation Article. In *Encyclopedia of Computer Science*.

Special Report Autonomous Vehicles. (2018). *The Economist*, 3–12.

SRI International. (n.d.). Shakey the Robot. Retrieved January 15, 2018, from https://www.sri.com/work/timeline-innovation/timeline.php?timeline=computing-digital#!&innovation=shakey-the-robot

Sun, Y., & Schonfeld, P. (2015). Stochastic capacity expansion models for airport facilities. *Transportation Research Part B: Methodological*, *80*, 1–18. https://doi.org/10.1016/j.trb.2015.06.009

The World Bank. (2017). Air Transport, Passengers carried. Retrieved from https://data.worldbank.org/indicator/IS.AIR.PSGR

U. Lima, P., & M. Custódio, L. (2005). Multi-Robot Systems (pp. 1–64). https://doi.org/10.1007/10992388_1

Vanderlande. (2017). FLEET: future-proofing baggage logistics. Retrieved November 16, 2017, from https://www.vanderlande.com/airports/innovative-systems/fleet

Vis, I. F. A. (2006). Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, *170*(3), 677–709. https://doi.org/10.1016/j.ejor.2004.09.020

Vollmann, T. E., Berry, W. L., & Whybark, D. C. (1988). *Manufacturing, Planning and Control Systems*. New York: Dow (Jones & Irwin).

Vosniakos, G.-C., & Mamalis, A. G. (1990). Automated Guided Vehicles System Design for FMS Applications, *30*(1), 85–97.

Wang, H.-F., & Chang, C.-M. (2015). Facility Layout for an Automated Guided Vehicle System. *Procedia Computer Science*, *55*(Itqm), 52–61. https://doi.org/10.1016/j.procs.2015.07.007

Weber, A., & Thomas, R. (2005). Key Performance Indicators - Measuring and Managing the Maintenance. *IAVARA Work Smart*, (November), 1–16. https://doi.org/10.1016/j.ecolind.2009.04.008

Wilson, D. L. (2005). Use of Modeling and Simulation to Support Airport Security. *Aerospace and Electronic Systems Magazine, IEEE*, *20*(8), 3–6. https://doi.org/10.1109/MAES.2005.1499286

Wooldridge, M. (2002). An Introduction to Multi-Agent Systems, 1–2.

Wu, P. P. Y., & Mengersen, K. (2013). A review of models and model usage scenarios for an airport complex system. *Transportation Research Part A: Policy and Practice*, *47*, 124–140. https://doi.org/10.1016/j.tra.2012.10.015

Wurman, P. R., D'Andrea, R., & Mountz, M. (2008). Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses. *AI Magazine*, *29*(1), 9. https://doi.org/10.1609/aimag.v29i1.2082

Wyss, G. D., & Jorgensen, K. H. (1998). A User's Guide to Sandia's Latin Hypercube Sampling Software. *SAND Report*, (July), 88. https://doi.org/98-0210

Xiang, X., Kennedy, R., & Madey, G. (2005). Verification and Validation of Agent-based Scientific Simulation Models. *Agent-Directed Simulation Conference*, 47–55.

Zhou, Y., Hu, H., Liu, Y., & Ding, Z. (2017). Collision and Deadlock Avoidance in Multirobot Systems: A Distributed Approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(7), 1712–1726. https://doi.org/10.1109/TSMC.2017.2670643

Zhu, K., Ludema, M. W., & Heijden van der, R. E. C. M. (2000). Air Cargo Transport by Multi-Agent Based Planning Delft University of Technology. *Proceedings of the 33rd Hawaii International Conference on System Sciences*, *0*(c), 1–10.

Appendix A – Scientific Summary

Autonomous Transport Robots in Baggage Handling Systems at Airports

A.A.E. Silven
Delft University of Technology
Delft, The Netherlands
alisasilven@gmail.com

***Abstract* – This scientific summary describes a research performed to investigate in which way it is feasible to dynamically alter the floor plan and desired capacity of airport baggage handling systems by making use of autonomous individual transport robots. By researching the state of the art of both conventional baggage handling systems and transport robot systems, a new baggage handling solution called the Baggage Robot Concept is proposed. A simulation model of this Baggage Robot Concept is developed to evaluate the performance of this concept in the sorting process of a baggage handling system at medium-sized regional airports. The research identified that the most important design elements for the Baggage Robot Concept are the number of robots and the floor layout configuration. The Baggage Robot Concept turns out to be a feasible concept. An important design choice that is identified in the research is the trade-off between manoeuvre space of robots and the location of charging and storage positions. This scientific summary summarizes a study that has been a first exploration in integrating autonomous robot systems in baggage handling systems, contributing to future proof and cost efficient operations.**

Keywords: Baggage Robot Concept; Autonomous Transport Robots; Baggage Handling Systems; Sorting Process; Agent-Based Modelling

I. INTRODUCTION

International air travel demand has been increasing for many years now and forecasts show that this growth will also be substantial in the future [1]. This implies challenges for airports worldwide as main airports have difficulties matching their current capacity with the ever-increasing air travel demand [2] [3]. At airports worldwide, elements such as runways, terminals, air traffic control and more need to be expanded to cope with the growing number of passengers [4]. One of the critical elements determining the capacity of an airport is the baggage handling system [5]. A traditional baggage handling system – or BHS – consists of a complex system of conveyor belts with many connected elements. The rigid nature of these conveyor belts make the system laborious and lengthy to implement, and also incapable of adapting to changing circumstances. This, in combination with the long-term uncertainty of demand, makes it impossible for the conventional baggage handling system to properly adapt to the demand fluctuations in the aviation industry. Currently, baggage handling systems are 'oversized by design': the systems have spare capacity to be able to handle growth in passenger numbers and to allow for failures in parts of the system. This prevents the need for alterations to the implementation as much as possible, but comes with an undesirable cost as a result of having overcapacity for many years or capacity that will never be used.

This leads to a need for research that focuses on the development of a new baggage handling concept at airports, to cope with these challenges and to be less dependent on the rigid conveyor belt systems. This new baggage handling concept must provide dynamic capacity and eliminate as much as possible the need to invest in overcapacity that will not be used for a number of years. This research investigates if a so-called Baggage Robot Concept – meaning using autonomous and individual transport robots in a baggage handling system – can help in making the floor plan and desired capacity of airport baggage handling systems more dynamic.

By investigating the state of the art of both baggage handling and transport robot systems, the Baggage Robot Concept is introduced, combining the two systems. An agent-based simulation model is then used to test the sorting process of this Baggage Robot Concept on a medium-sized regional airport operating in a point-to-point network.

II. RESEARCH APPROACH

The situation of growing air travel demand and its implications show the urge of dynamically alter capacity of baggage handling systems. The goal of this research is to investigate the use of autonomous and individual transport robots in baggage handling systems. The research is structured by following the design science methodology as formulated by Peffers et al. (2007) [6].

Conventional baggage handling systems are 'oversized by design' to have spare capacity to be able to handle growth in passenger numbers in the future and to allow for failures in parts of the system. The motivation to research alternative systems that enable dynamic capacity adjustments is the undesirable cost as a result of having this overcapacity for many years. The 2050+ Airport research project, supported by the European Commission, underlines this motivation by proposing to use small transport robots [7]. By researching the state of the art of both conventional baggage handling systems and transport robot systems, the most important elements of the Baggage Robot Concept are identified. Next to that, key performance indicators useful for assessing the performance of this new concept, as well as the identification of requirements, both functional and non-functional are identified. These system requirements and key performance indicators are then used to develop a simulation model of a part of the Baggage Robot Concept. By using this model, the Baggage Robot Concept is demonstrated on a small scale. This demonstration can serve multiple purposes, from proving that the Baggage Robot Concept works to a more formal evaluation of the simulation model itself.

Finally, the simulation model is used to experiment with both the model and the Baggage Robot Concept itself. By experimenting with the model, the effects of different input parameter values on the performance of the model and the Baggage Robot Concept can be tested by using different experimental designs. By varying for example the number of robots and the floor layout configuration, the effect of the different input parameters on the performance of the Baggage Robot Concept can be identified. By using the key performance indicators formulated, an evaluation of the Baggage Robot Concept itself can be performed. From running experiments and generating performance data from the simulation model, observations and measures can show to what extent the autonomous and individual transport robots provide a solution to the research problem.

III. THE BAGGAGE ROBOT CONCEPT

The main purpose of a baggage handling system in an airport is to transport checked bags from bag drop off facilities to makeup stations, where the bags are loaded on baggage carts that bring them to the aircraft. The Baggage Robot Concept serves the same purpose. By using individual transport robots that autonomously and real-time decide on their preferred paths, bags are transported between the drop off facilities and makeup stations, replacing the complex system of conveyor belts that is currently used. Fixed machines such as the different security screening machines can stay at their original location and the transport robots are used to transport bags to, between and from these machines. The rough sketch in Figure 1 shows the difference between a conventional baggage handling system (left) and the Baggage Robot Concept (right). The conveyor belts in the security screening and sorting process are removed. The orange lines indicate the shortest paths between the security screening layers and three makeup stations. These orange lines do not indicate paths that the robots *have* to take, as robots can freely move around in the baggage handling area. The lines rather indicate possible paths. The black and orange squares in the right sketch of the Baggage Robot Concept indicate the individual transport robots moving in the area.

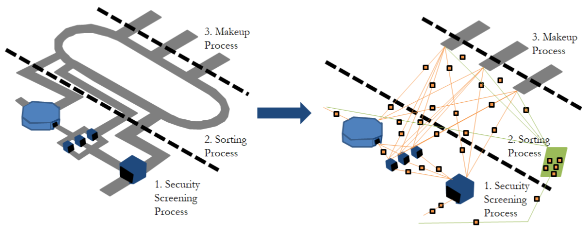

Figure 1 – Conventional BHS versus the Baggage Robot Concept – replacing conveyor belts by transport robots

The robots can bring bags to and pick up bags from the security screening machines by placing the bags on a small conveyor belt that goes through the security screening machine and pick it up after the screening is completed. The robots do not go through the machines themselves. Once the bags are cleared in the security screening process, they continue to the sorting process.

During the sorting process in a baggage handling system, bags need to be transported from the area where bags are security checked to one of the makeup stations in the baggage handling facility. By using individual transport robots instead of conveyor belts, bags can be transported to the correct makeup station directly. The robots fulfil the transport and sorting task simultaneously, taking the bag straight to the correct makeup station.

As each individual robot has a battery, a charging facility is needed to charge these batteries. The battery charging is done by inductive charging. Inductive charging positions can be anywhere in the area. The sketch in Figure 1 indicates a charging area (in green) with multiple charging positions. The green lines indicate paths that robots can but do not have to take to arrive at these charging positions. The charging area serves a double function as it can also function as storage area. The same drip-feeding technology of AGVs can be applied here. Robots can be drip-fed while standing idle in the charging and storing area, making sure the battery is continuously charged during idleness. Robots that have completed a transport job, i.e. have picked up a bag from a bag drop facility and unloaded the bag at the correct makeup station, can take on a new transport job as long as their battery is sufficiently charged to complete another transport job.

Design Elements of the Baggage Robot Concept

In this section, the most important design elements of the Baggage Robot Concept are described.

Layout Configuration

The layout of a baggage handling area in which individual transport robots transport bags from the drop off point to a makeup station can be adjusted easily. The robots do not depend on fixed infrastructure; they only need a charging station. By integrating electromagnetic induction in the floor of the baggage handling area, the charging infrastructure does not result in any obstacles for transport robots in motion. The location of charging positions can differ per baggage handling area, depending on the original layout of the area when a baggage robot system is integrated in an existing baggage handling area or can be decided upon during the design phase of a new baggage handling area.

Control

One of the most important considerations in the design of a new transport robot system is the control architecture, in particular the technique used to coordinate the motions of the individual vehicles [8]. Robots in the Baggage Robot Concept are considered to be autonomous. They have some control over their state and behaviour and are able to react to actions of robots in their proximity. A hybrid control architecture provides suitable control of the Baggage Robot Concept. It increases the robustness, scalability and performance of the system.

Routing

From the perspective of passengers, it is important that their bags are loaded into the correct aircraft and in time. To create as much slack time as possible in the sorting and other processes of baggage handling, the preferred path bags travel in the baggage handling area is the shortest path. This means that bags have to be transported between m entrances and n exits of the baggage handling system, taking the shortest path between these points. As robots do not exit the sorting system when they unload a bag at a makeup station, they also need paths or a route back to one of the m entrances to pick up new bags. These return trips can be interrupted when the robot runs low on battery power and needs to charge its battery to continue its path to one of the m entrances. This storage and charging area can also be used by robots to stand idle and drip-feed their battery when there are no bags incoming.

Collision and Deadlock Avoidance

When a number of robots have to perform their transport tasks in the same area, as is the case in this Baggage Robot Concept, there is a risk of collisions. In this concept, robots plan their paths independently. The path they initially plan is collision free, but there is no guarantee that this path will remain collision free as all robots in the system are able to change their own paths at any time. This may result in a situation in which robots have already started moving along their path when they detect another robot getting close to crossing their path [9]. This detection of other robots in their proximity is one of the most important abilities of the robots to realize collision avoidance. The most effective way to detect the other robots close by is by obtaining information on the planned paths of these closest robots by means of local communication [10]. To avoid a collision between the robots, re-planning of one of these paths is necessary to guarantee collision free paths for both robots involved [9]. The robots involved exchange information on their positions and planned paths. Together they need to decide who gets permission to go and who has to give way. To make this decision, predetermined traffic rules can be used that depend on the type of predicted collision.

Two types of collisions for multi-robot systems as distinguished by Mauro (2017) are side collisions and frontal collisions [11]. In case of an imminent side collision, stopping and resuming policies are used to avoid actual side collisions. Robots transporting bags have priority over empty robots in collision avoidance situations. When two robots with the same priority are threatened by a side collision, priority cannot be decisive and one of the robots will be randomly picked to move first. This random picking can be done by the centralized component or the two robots negotiate who goes first by for example "pulling straws" and assigning a winner based on centrally defined rules.

In case of a possible frontal collision, a more relaxed version of Mauro's (2017) right turn policy is proposed, being a turn, (wait) and continue policy. Both robots stop moving forward and instead move one step to a neighbouring position that is unoccupied and then re-calculate and resume their new shortest path. If all neighbouring positions are occupied, robots wait until one of these neighbouring positions is vacated again, after which it will resume a shortest path. A right-moving policy is considered but found to be too constraining as when the position on the right is occupied, deadlock is lurking.

To determine when a collision avoidance measure should be invoked, the Baggage Robot Concept uses a safety zone for each robot. This safety zone can also be referred to as a safety distance d_{safe} between two robots. When the distance between two robots is larger than or equal to the safety distance, no possible collision will occur in the near future. When this distance becomes smaller and the safety zones start to (partially) overlap, one of the collision avoidance measures is invoked to avoid an actual collision, depending on whether the possible imminent collision is a side collision or a frontal collision.

By invoking the collision avoidance measures discussed when collisions are imminent, deadlocks are avoided too. As the turn and continue policy is a relaxed version of the right turn policy, robots have more neighbouring positions to go to. By adding the possibility to wait one or more time steps in this policy, unsolvable deadlocks are avoided.

This form of hybrid control for collision avoidance uses centralized components to achieve global coordination by decentralized algorithms and assumes only local communication between pairs of physically close robots. The local communication between the robots allows for more adaptive coordination between the robots when planning their paths, supported by a centralized component [12].

Key Performance Indicators

Key performance indicators are necessary to evaluate the performance of the Baggage Robot Concept. When this new system performs as well or even better than the conventional version, the feasibility is proven. At the same time, the new system provides more flexibility in capacity by using individual and autonomous transport robots. Key performance indicators that are relevant in measuring the performance of the new sorting system for baggage handling are indicated in Figure 2.

These KPIs can be prioritized dependent on stakeholder views. Stakeholders include the owner of the baggage handling system (airport and/or ground handler), airlines and passengers. The owner of the baggage handling system is interested in all KPIs, whereas the airlines and passengers are only (in)directly interested in the average process time of bags and the percentage of bags exceeding the norm time as these KPIs are related to mishandled bags. In case of mishandled bags, passengers are duped and airlines have to compensate these passengers.

	Key Performance Indicators	Unit	Optimal value
Sorting System	• Average process time of bags	[time]	6 minutes
	• Percentage of bags exceeding norm time	[%]	0%-0.02%
	• Average number of robots	[#]	As low as possible
	• Occupancy rate		
	– Percentage of operational time while being loaded	[%]	As high as possible
	– Percentage of operational but empty trips	[%]	A value as close to 50% as possible
	• Percentage of charging time	[%]	As low as possible
	• Number of conflicts avoided	[#]	As low as possible

Figure 2 - Key Performance Indicators of the sorting process of the Baggage Robot Concept

To research the relation between design elements key performance indicators, a simulation model of the Baggage Robot Concept has been constructed. The next section elaborates on the simulation model used to evaluate the Baggage Robot Concept.

IV. SIMULATION MODEL

Constructing a real-size test setup of the Baggage Robot Concept is laborious and expensive. Using modelling and simulation a low-cost, less time-intensive evaluation of a part of the rough design of the Baggage Robot Concept as described in the previous section can be performed. The method of choice in this research is Agent-Based Modelling (ABM). The most important reason for this choice is the autonomous nature of the individual transport robots considered. In literature, the single most given reason on using ABM can be paraphrased as that agent-based model are able to explicitly model the complexity that arises from individual actions and interactions that arise in the real world [13].

The goal of using the simulation model to run experiments is to provide the user of the model with insights into the performance of a baggage handling system that uses autonomous and individual transport robots for the sorting job of the system. The model is constructed using modelling steps as derived from Marion, Scotland, Lawson, & Marion (2008) and Maki & Thompson (2005), roughly consisting of: conceptualisation, model implementation, model verification and validation, model experimentation and analysis of experiment results [14] [15].

The scope of the model is chosen to be the sorting process in the baggage handling system. It is assumed that the biggest lead time improvements can be achieved in the sorting system. The model requirements are to minimize the bag waiting time by positioning the charging and storage areas as close to the incoming conveyor belts as possible and at the same time to maximize the reachability of these positions by giving each position at least one free neighbour position for robots to move over. Furthermore, basic constraints and assumptions simplify the simulation model.

Model Elements

Similar to the description of the Baggage Robot Concept in section III, the most important model elements are discussed.

Layout Configuration

The baggage handling area in this research is considered to have fixed dimensions. The location of basic elements such as incoming conveyor belts and entrances to makeup stations are fixed as well. However, the locations where individual transport robots are stored and simultaneously charged by means of drip-feeding can be altered. This research considered four layout options which are depicted in Figure 3. Green squares indicate the location of the charging and storage area. The layout configurations are named after these locations. The top left configuration is called 'central areas', the top right configuration 'six vertical areas', bottom left is called 'four horizontal areas' and bottom right 'six horizontal areas'.

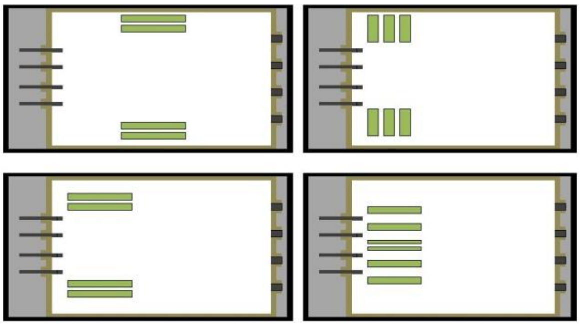

Figure 3 - Four Layout Configurations

These layout configurations are not the only layouts imaginable but rather are designed to demonstrate how the location of storage and charging areas affect the performance of the system as a whole. Corresponding to the first design requirement – minimizing bag waiting time by locating charging and storage areas as close to the incoming conveyor belts as possible – layouts 2 to 4 are chosen. The first layout 'central areas' serves as a comparison, to check whether or not positioning the storage and charging areas as close to the incoming conveyor belts as possible has an influence on the waiting time of bags, opposed to the layout where the storage and charging positions are not located as close to the belts as possible.

Control

The preferred control approach for the Baggage Robot Concept as a whole is hybrid control. The simulation model that focuses on the sorting process of the Baggage Robot Concept incorporates this type of control.

When sensors detect a bag at the end of an incoming conveyor belt, it sends out a signal to a central control unit. This central control unit assigns the closest available and sufficiently charged robot to the transportation task of transporting the detected bag to a makeup station (centralized control). Once the robot received the task, it independently performs the transportation task, while continuously being aware of its environment (decentralized control). While executing the transportation tasks, robots have to obey to the 'traffic rules' that are centrally defined.

Once the robot delivered the bag, it drives back to a free storage and charging position and when it is charged enough, it sends out a signal to the central control unit that it is available again and ready to receive a new transportation task. This combination of centralized and decentralized control components in the model shows how the hybrid control approach is implemented in the model.

Routing

The problem considered is the transportation of bags between a set of fixed incoming conveyor belts O (origin) and fixed entrances to makeup stations D (destination). For each incoming bag a robot has to be assigned to the transportation task of the incoming bag to transport the bag from its incoming conveyor belt m ∈ O to the makeup station that corresponds with the destination the bag needs to be sorted to n ∈ D. The transportation task or sorting of bags to a makeup station can be defined as $S^{mn} \in S(t)$.

For each individual transport robot the shortest route from the origin to the destination needs to be calculated. These shortest paths are therefore defined from the point of view of the individual robots. The objective of this problem is the following:

$$minimize \sum_{ij \epsilon A} d_{ij}$$

Subject to

$$d \geq 0$$

$$\sum_j d_{ij} - \sum_j d_{ji} = \begin{cases} 1 & if\ i = O \\ -1 & if\ i = D \\ 0 & otherwise \end{cases} \quad \forall\, i$$

In which the sorting area of the baggage handling system is considered to be an undirected graph (V, A) with source node O, destination node D and cost expressed in distance d_{ij} for each edge (i, j) in A. To find this shortest path, the A* algorithm is used. A* is found to be a faster version of the well-known Dijkstra algorithm as it is a best-first search algorithm. As the Dijkstra algorithm requires more computational power at every step in the simulation model, A* is found to be better suitable for simulation purposes.

Collision and Deadlock Avoidance

To avoid a deadlock situation, robots have the ability to communicate with other robots in their direct surrounding and one step further. When two robots are in each other's proximity and want to go to the same position at the same time, they communicate before taking a step. In this communication step, they exchange information on where they want to go and what their priority is. This thinking ahead mechanism of the robots make sure that not more than one robot can take the actual step to a position that is desired by more than one robot. This avoids collisions, but also deadlocks, as robots are able to think not only one but two steps ahead.

Model Verification and Validation

The model is verified using four types of verification tests as suggested by Van Dam et al. (2010) [16]. The verification tests performed do not entirely complete the model verification. As there are an infinite number of input parameter combinations and variations possible, the verification of this agent-based model is never complete. By performing multiple different tests for the four different main parts of verifying agent-based models as proposed by Dam et al. (2010), an effort is made to gain a sufficient confidence in the developed simulation model.

To validate the agent-based model, the method of face validation through expert validation is used. Three experts face validated the developed model with respect to the model purpose. They acknowledged that conventional baggage handling systems have the main disadvantage that future extensions have to be taken into account from day one as especially sorter equipment is not easy to modify when it is in live operation. They argue that replacing these conveyor belts by individual transport robots can provide more flexibility. Although experts can be wrong too, Dam et al. (2010) argue that this method is still an appropriate way to address agent-based model validation, meaning that a model that is face validated by experts can be considered good enough.

V. RESULTS

The verified and face validated model can be used to perform experiments. By performing experiments, various strategies for the operation of the system can be evaluated [13].

Experimental Setup
For this research, three experiments are set up. In the first experiment, these input parameter values are set to represent presumed 'normal conditions'. Normal conditions are considered to be the conditions on an average day in the year so no unexpected disruptions like dysfunctional systems or heavy snow. By fixing the parameters, the performance of Baggage Robot Concept in the sorting part of a baggage handling system on a medium-sized regional airport can be examined.

The second experiment however, requires all input parameters to vary simultaneously. The input parameters are:

- Layout configuration
- Number of robots
- Battery reduction rate when robot is idle
- Battery reduction rate, robot empty, driving
- Battery reduction rate, robot loaded, driving
- Battery charging rate of robot
- Battery level threshold

The values of these input parameters need to be established to be able to run multiple different experiments in a smart way. Latin Hypercube Sampling is used to run experiments with smart variations in the input parameters. This second experiment tries to indicate which input parameters affect the output metrics the most for two scenarios; 1624 bags are inserted into the system and 4290 bags are inserted, representing an extreme case.

The third experiment investigates the most influential input parameters further. By only varying the number of robots and the layout of the manoeuvre area of the robots and keeping the other input parameters constant, the effect of these input parameters can be demonstrated with more precision. This experiment is also executed for both number of bags scenarios.

Experimental Results

Experiment Design I: Fixed Settings
Figure 4 shows four boxplots for four KPIs: the average process time of bags, the percentage charging time, the number of avoided conflicts and the percentage of mishandled bags. KPIs on the robot utilization rates are not shown as they are all around the 50%-50% ratio and the number of robots is used as an input parameter.

The results of the first experiment design show that this combination of input parameter values is able to meet the average process time of bags KPI. The average process time of bags is far below the maximum allowed threshold of 6 minutes or 360 seconds. Even when unexpected minor disruptions become present, there is enough slack time to avoid significant problems. The layout configuration chosen in this experiment design can cause a minor detour for the robots as they can choose to which charging and storage position they want to go.

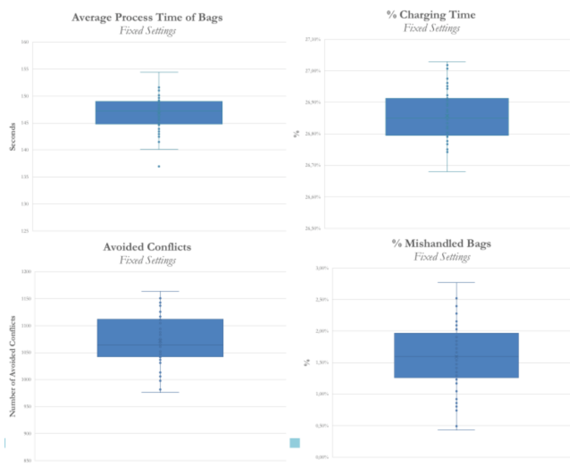

Figure 4 - Results Experiment Design I

This experiment shows a ratio of 51.92%-48.08% for empty and loaded trips, which is considered within the ranges of being optimal. The percentage of charging time should be as low as possible while ensuring all bags are handled, which is the case in this experiment. A value of 26.68% for 8 robots is considered fine, indicating a good balance between the number of robots and the number of incoming bags over time. The number of avoided conflicts cannot be judged to be low or high. This combination of input parameter values leads to a situation in which the percentage of mishandled bags is higher than 0.02% in all runs. This means the selected set of input parameters is not able to comply with all the theoretic optimal values for the KPIs, showing that the base case values chosen do not fulfil the practical requirements to the Baggage Robot Concept. The next experiment shall provide more insight into which combination of design parameter values actually provides a feasible design.

Experiment Design II: Parameter Sweep
This experiment design is performed under two scenarios: inserting 1624 bags into the system and inserting 4290 bags into the system, representing an extreme scenario. Figure 5

shows the boxplots of four KPIs for both scenarios for the second experiment design.

In the 1624 bags scenario, the combination that has the lowest value on the average process time of bags KPI combines 53 robots with the six horizontal areas configuration. This implies that if more than 53 robots are present, they get in each other's way while moving around, resulting in more conflicts to be avoided, resulting in more time the robots have to wait for each other, resulting in a higher average process time for bags. Considering the number of avoided conflicts, this experiment shows that the central areas configuration results in the most optimal value for this KPI, as there is more manoeuvre space for the robots than the other layout configurations. This demonstrably results in less avoided conflicts.

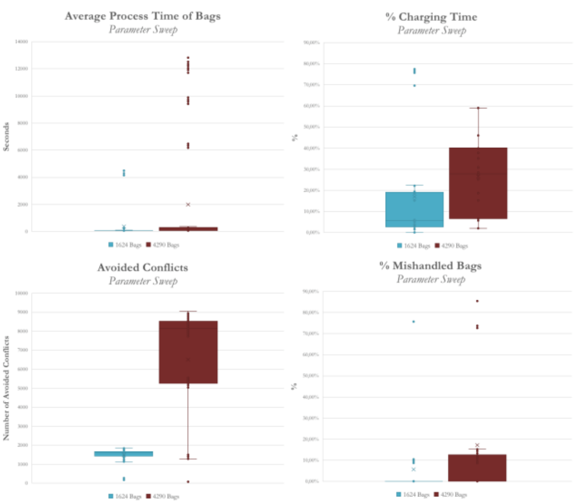

Figure 5 - Results Experiment Design II

For the 4290 scenario, all obtained values comply with the theoretic optimal value for the KPIs as stated. Compared to the 1624 scenario, the average process time of bags is 26.25% higher – still easily complying with the maximum threshold of 360 seconds – while the number of bags to be transported is 164.16% higher. The minimum number of robots necessary to handle all these 4290 bags is however higher, 26 compared to 15. When 4290 bags need to be handled, the percentage of empty trips is slightly higher compared to the 1624 bags situation, which can be explained by the fact that the combination of input parameter values causing the optimal value for this KPI includes having 57 robots. It is expected that at certain moments most of these robots are operational and as there are quite many, they might run into each other a lot, resulting in more conflicts to be avoided, which results in more waiting time especially for empty robots as loaded robots get priority over empty robots, increasing the percentage of empty trips in time. The same logic applies to the percentage of loaded trips. The optimal value for charging time is higher than in the 1624 bags scenario. In the 1624 bags scenario the optimal value for this KPI is obtained by having 76 robots, while in the 4290 bags scenario this value is obtained by a combination of input parameter values that include having 51 robots. A fewer number of robots have to transfer significantly more bags, resulting in more need for charging and thus a higher percentage of charging time. The number of avoided conflicts is also significantly higher in the 4290 bags scenario compared to the 1624 bags scenario, due to the number of robots present in the combination that results in the optimal values; 19 robots for 1126 avoided conflicts in the 1624 bags scenario compared to 34 robots for 7733 avoided conflicts in the 4290 bags scenario.

The next experiment design examines the effects of the input parameters 'number of robots' and 'configuration' more closely as they showed to have the greatest impact on the KPIs. Both the 1624 and the 4290 bags scenario are examined for the different combinations of input parameter values for these two input parameters.

Experiment Design III: Influential Parameters

This experiment tries to show the minimum required number of robots per layout configuration, to comply with all the optimal KPI when the other input parameters such as battery charging rate and the battery level threshold are kept constant considered to represent 'normal conditions'. Figure 6 shows histograms of the 1624 bags scenario. For the 4290 bags scenario, the histograms show the same trends in course of the results.

The results show that the more robots are used, the lower the average process time of bags becomes. However, after a certain point, the addition of an extra robot does not contribute a great deal to the reduction in average process time of bags. For the 1624 bags scenario, having 9 robots causes the average process time to drop below the set threshold for all configurations, for the 4290 bags scenario 25 robots are needed. Adding an extra robot does decrease the average process time but cost estimations for the purchase of an additional robot have to show if it is worth it. Considering the average percentage of time that robots are charging, a downwards trend starting from approximately 10 robots in each configuration for the 1624 bags scenario is detected and for the 4290 bags scenario from around 22 robots.

The bottom left histogram shows the number of avoided conflicts per configuration. From this upwards trend it can be concluded that robots get in each other's way more when the number of robot increases. In this histogram however, there is a clear difference visible between the configurations. This is true for both scenarios. The central areas configuration shows a significantly lower number of potential conflicts than the other configurations, regardless of the number of robots present. This can be explained by the manoeuvre space that this layout causes. As the charging and storage positions are located at the top and bottom of the layout as seen from above, robots have a larger open space to move around. The other areas have the same dimensions but are located in such a way that the robots have less manoeuvre space near the incoming conveyor belts, resulting in different shortest path possibilities.

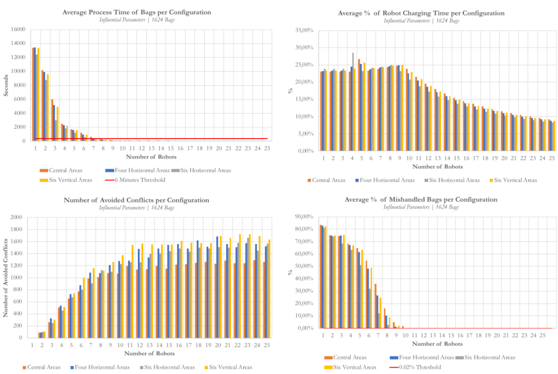

Figure 6 - Results Experiment Design III

The bottom right histogram shows the average percentage of mishandled bags per configuration. A clear downward trend is visible here and from 11 robots onwards, the value for this KPI drops below the 0.02% threshold. Adding an additional robot does not decrease this percentage even more, as it reaches a value of 0.00% in all configurations with 11 robots for the 1624 bags scenario. For the 4290 bags scenario, the central area configuration is not able to meet the percentage of mishandled bags KPI threshold, not even with 96 robots, which is the maximum number of robots tested. For the other three layout configurations, 21 robots are enough to have the percentage of mishandled bags drop below the threshold of 0.02%.

VI. CONCLUSION & FUTURE RESEARCH

By implementing the Baggage Robot Concept, a step is made in dynamically altering the floor plan and desired capacity of airport baggage handling systems. The performance of this Baggage Robot Concept has been reflected upon using the simulation results. Exact design parameters or values for these parameters cannot yet be given due to the limitations of the developed simulation model and the lack of reliable data available for a far-future concept. Yet, the number of robots and the layout configuration were found to be the most important design parameters. The number of robots should be such that the average process time of bags does not significantly decrease when introducing an additional robot to the system. The layout should be such that the route the robots travel is as short as possible, while at the same time ensuring the robots have enough manoeuvre space to reduce the number of possible conflicts. An example of such a layout that was chosen in this research is the Six Horizontal Areas layout, in which the charging and storage location of robots is located close to the incoming conveyor belts. Depending on the number of bags to be handled and the arrival pattern of these bags, a different minimum number of robots is necessary. When taking all the KPIs into account, this number should be higher (because of the determining KPI 'percentage of mishandled bags') then when the other KPIs don't play a role and the acquisition costs of robots are the main determinant. Depending on the interest of the system owner, the importance of the KPIs can shift.

This thesis has been a first exploration in integrating autonomous robot systems in baggage handling systems, contributing to future proof and cost efficient airport operations.

Future Research

Future research should investigate the success of the Baggage Robot Concept as a supplement to or a replacement of conventional baggage handling systems. A suggestion for future research is to explore the technical feasibility of the proposed robot types. One important starting point can be to research the sensing capabilities of the transport robots to gain insight into the technical possibilities of these robots. To do so, inspiration can be taken and lessons can be learned from innovations in the field of autonomous cars. Although the autonomous car technology is still being developed, it can help to improve the individual transport robot systems. In this future research, timescale and cost estimations for research and development can help in estimating on which timescale the proposed Baggage Robot Concept could be built into a prototype system to further test it.

During the modelling process, scoping choices have been made resulting in a heavy NetLogo model of the first high-level design of the sorting process of the Baggage Robot Concept. To give a full picture on the performance of the Baggage Robot Concept in the broadest sense, future research can extend the scope of the model to the security screening and the makeup process, the other two major processes in baggage handling. To extend this future research, implementing and experimenting with more incoming conveyor belts and makeup stations, as well as the locations of these facilities in the baggage handling area can add value to the existing model.

If in future research the concept is brought further, more detailed simulation methods should be used to prove more detailed results and evaluative hypotheses about the workings of the detailed design. To thoroughly validating the model, several of the validation methods mentioned in section IV can be used to not only give the agent-based simulation model face validity, but fully validate it.

ACKNOWLEDGEMENTS

The author acknowledges that the work presented in this paper has extensively benefited from discussions with multiple advisors at Delft University of Technology and NLR, including prof. dr. R.R. Negenborn, dr. J.H.R. van Duin, dr. ir. Y. Pang and ir. R.J.D. Verbeek.

REFERENCES

[1] Graham, A., & Metz, D. (2017). Limits to air travel growth : The case of infrequent flyers. *Journal of Air Transport Management*, *62*, 109–120.

[2] Mota, M. M., & Boosten, G. (2011). Extended Definition of Capacity in Airport Systems. In *ISNGI*.

[3] Sun, Y., & Schonfeld, P. (2015). Stochastic capacity expansion models for airport facilities. *Transportation Research Part B: Methodological*, *80*, 1–18.

[4] IATA. (2016). IATA Forecasts Passenger Demand to Double Over 20 Years. Retrieved September 26, 2017

[5] Cavada, J. P., Cortés, C. E., & Rey, P. A. (2017). A simulation approach to modelling baggage handling systems at an international airport. *Simulation Modelling Practice and Theory*, *75*, 146–164.

[6] Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77.

[7] European Commission. (n.d.). 2050AP Report Summary.

[8] Mas, I., & Kitts, C. (2010). Centralized and decentralized multi-robot control methods using the cluster space control framework. *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, (August), 115–122.

[9] Zhou, Y., Hu, H., Liu, Y., & Ding, Z. (2017). Collision and Deadlock Avoidance in Multirobot Systems: A Distributed Approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(7), 1712–1726.

[10] Arai, Y. (1999). Collision avoidance in multi-robot systems based on multi-layered reinforcement learning. *Robotics and Autonomous Systems*, *29*(1), 21–32.

[11] Mauro, F. (2017). Towards the design of an effective and robust multi-robot parcel sorting system. Delft University of Technolgy.

[12] Jäger, M., & Nebel, B. (2001). Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. IEEE International Conference on Intelligent Robots and Systems, 3, 1213–1219.

[13] Siebers, P. O., Macal, C. M., Garnett, J., Buxton, D., & Pidd, M. (2010). Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, *4*(3), 204–210.

[14] Marion, G., Scotland, S., Lawson, D., & Marion, G. (2008). An Introduction to Mathematical Modelling List of Figures.

[15] Maki, D., & Thompson, M. (2005). *Mathematical Modeling and Computer Simulation* (1st ed.). Cengage Learning.

[16] Dam, K. H. van, Nikolic, I., & Lukszo, Z. (2010). *Agent-Based Social Systems Volume 7 Editor in Chief: Series Edirotrs:* (H. Deguchi, Ed.) (Vol. 9). Springer.

Appendix B – Stakeholders Involved in the Baggage Handling Process

Table B 1 - Stakeholders Involved in the Baggage Handling Process

Actor	Interests	Desired situation/objectives	Existing or expected situation and gap	Causes	Possible solutions
Airlines	Minimize the costs for compensation in case of lost or delayed bags	Have a significant market share and close to zero expenses on compensations for lost or delayed bags	Bags get lost or delayed in the baggage handling system of an airport, resulting in the airline having to give the affected passenger(s) a compensation	The conventional baggage handling system with conveyor belts is not 100% flawless	Having a baggage handling system that reduces the chances of bags getting mishandled or delayed
Airports	Have or keep a good reputation on baggage handling to attract passengers	To become the most important airport in the region	Bags get lost, delayed or wrongfully sorted in the baggage handling system, affecting the reputation of the airport in the eyes of passengers	The conventional baggage handling system with conveyor belts is not 100% flawless	Having a baggage handling system that reduces the chances of bags getting mishandled or delayed
Ground handlers	Win and maintain tenders by guaranteeing an outstanding performance at minimum costs	Carry out all the baggage handlings at an airport with outstanding performance and minimum costs to win and maintain tenders	Bags get lost, delayed or wrongfully sorted, affecting the performance quality and reputation of the ground handler	The conventional baggage handling system with conveyor belts is not capable of dynamically altering capacity and is not 100% flawless	Having a baggage handling system that has dynamic capacity to minimize operational costs and can guarantee accurate performance simultaneously
Passengers	Having the baggage loaded into the same aircraft as the passenger in time	Having the right baggage item(s) at the right reclaim belt at the right destination at the right time	Bags get lost, delayed or wrongfully sorted, resulting in passengers missing their baggage item(s) once they arrive at the destination airport	The conventional baggage handling system with conveyor belts is not 100% flawless	Having a baggage handling system that reduces the chances of bags getting mishandled or delayed

Appendix C – Path Lengths and Charging Calculations

C.1. Path Length Values per Layout Configuration

This section shows the values for the shortest and longest shortest paths between the charging and storage positions and the incoming conveyor belts for all four layout configurations. In the simulation model, robots move using a 'stepwise approach', meaning that robots cannot go to a position that is located diagonally from the robots current position in one step. For diagonal movements it needs two steps, opposed to a direct approach where diagonal movements can be done in one step. This difference is displayed in Figure C 1.

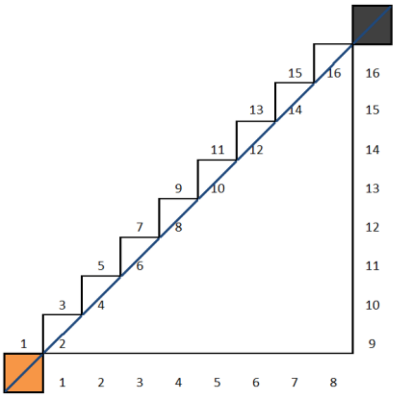

Figure C 1 - Stepwise approach (black lines) versus direct approach (blue line)

In the stepwise approach, multiple displays of the shortest path between two positions are possible. However, the value of the shortest path between two positions remains the same, regardless through which exact other positions it goes, as long as the start and end position remain the same and the shortest path algorithm is followed. Note that not all possible shortest paths in the stepwise approach are shown but just two to demonstrate that the lengths remain the same if the start and end node aren't changed and the shortest path logic is followed.

The tables below show the values for the shortest and the longest shortest path for all four layouts and for both the stepwise and the direct approach. The stepwise approach value is the first value, followed by an arrow indicating the difference between the stepwise and the direct approach. For example the shortest path from the closest charging and storage position to the closest incoming conveyor belt in the central areas layout configuration is 18 steps when the stepwise approach is used, which is the approach used. In further research one might want to check the effect of a different approach like the direct approach on the performance of the system. That is why the value for the shortest and longest shortest path for each layout configuration is also calculated for when the direct approach were to be used. As becomes apparent from the values in the tables below, the direct approach causes robots to get to the incoming conveyor belts in fewer steps for all layout configurations.

Table C 1 - Path Length Values Central Areas

Configuration 1: Central Areas Maximum number of robots: 96			
	# Steps		# Steps
Shortest path to belt 1	**18 → 12**	Longest shortest path to belt 1	**33 → 23**
Shortest path to belt 2	**23 → 12**	Longest shortest path to belt 2	**38 → 23**
Shortest path to belt 3	**29 → 17**	Longest shortest path to belt 3	**44 →23**
Shortest path to belt 4	**34 → 22**	Longest shortest path to belt 4	**49 → 26**

Table C 2 - Path Length Values Six Vertical Areas

Configuration 2: Six Vertical Areas Maximum number of robots: 96			
	# Steps		# Steps
Shortest path to belt 1	**5 → 3**	Longest shortest path to belt 1	**19 →10**
Shortest path to belt 2	**10 → 8**	Longest shortest path to belt 2	**24 → 15**
Shortest path to belt 3	**16 →14**	Longest shortest path to belt 3	**30 → 21**
Shortest path to belt 4	**21 → 19**	Longest shortest path to belt 4	**35 → 26**

Table C 3 - Path Length Values Four Horizontal Areas

Configuration 3: Four Horizontal Areas Maximum number of robots: 96			
	# Steps		# Steps
Shortest path to belt 1	**5 → 3**	Longest shortest path to belt 1	**20 → 13**
Shortest path to belt 2	**10 → 8**	Longest shortest path to belt 2	**25 → 18**
Shortest path to belt 3	**16 → 14**	Longest shortest path to belt 3	**31 →24**
Shortest path to belt 4	**21 → 19**	Longest shortest path to belt 4	**36 → 29**

Table C 4 - Path Length Values Six Horizontal Areas

Configuration 4: Six Horizontal Areas Maximum number of robots: 96			
	# Steps		# Steps
Shortest path to belt 1	**4 → 2**	Longest shortest path to belt 1	**15 →12**
Shortest path to belt 2	**9 → 7**	Longest shortest path to belt 2	**20 → 12**
Shortest path to belt 3	**15 → 13**	Longest shortest path to belt 3	**26 →14**
Shortest path to belt 4	**20 → 18**	Longest shortest path to belt 4	**31 → 19**

C.2. Charging Calculations

A robot should be fully charged within 4 hours which means that the battery percentage needs to go from 0 to 100% in 4 hours (14.400 seconds), which means that every second the battery charges 0.00694444 percentage points per second.

As it takes 4 hours to fully charge a battery and it is preferred to load an unused robot maximum once a day, the battery should last for at least 20 hours when the robot is unused. When the robot is idle for a while, it would proceed to the charging area, where the robot's battery is being drip-fed. Combined with a threshold of a battery percentage of 0.8%, a situation where a robot's battery reaches 0% will be avoided. To still have a safety margin, it is assumed that it takes at least 20 hours for a robot's battery to go from 100% to 0%, meaning the drain rate is assumed to be 0.001388889 percentage points per second.

Furthermore it is assumed that driving without a load requires three times as much battery as standing idle, corresponding to 3 x 0.001388889 = 0.00416667 percentage points per second. Driving around the baggage handling area with a bag on the robot is assumed to consume four times as much battery as standing idle, corresponding to 4 x 0.001388889 = 0.00555556 percentage points per second.

A threshold is set on the battery percentage for a robot to be allowed to start a new assignment. The longest path that can be taken by a robot consists of 157 patches, corresponding to 471 meters.

In the configuration with the longest shortest paths – configuration 1 – the shortest path from the furthest charging position to the furthest conveyor belt consists of 49 patches. The longest shortest path from the conveyor belt to the furthest makeup station is 55 patches. The longest shortest path from this makeup station to the furthest charging and storing position is 53. Adding the values lead to the longest shortest path to complete one transportation job for a robot. For the stepwise approach this cycle which is completed by following the longest shortest paths has a value of 157 patches. Figure C 2 shows these paths for the stepwise approach and the difference in case a direct approach was to be used. The red arrows show the path of the stepwise approach from the furthest charging and storage position to the furthest conveyor belt and from this conveyor belt to the furthest makeup station. The black arrows show the way back from this makeup station to the furthest charging and storage position. Green arrows indicate the path of the direct approach from the furthest charging and storage position to the furthest conveyor belt and from there to the furthest makeup station. Purple arrows indicate the path from this makeup station to the furthest charging and storage position. Note that the layout is symmetric; multiple longest shortest paths are possible, all leading to the same values for longest shortest paths forming a cycle.

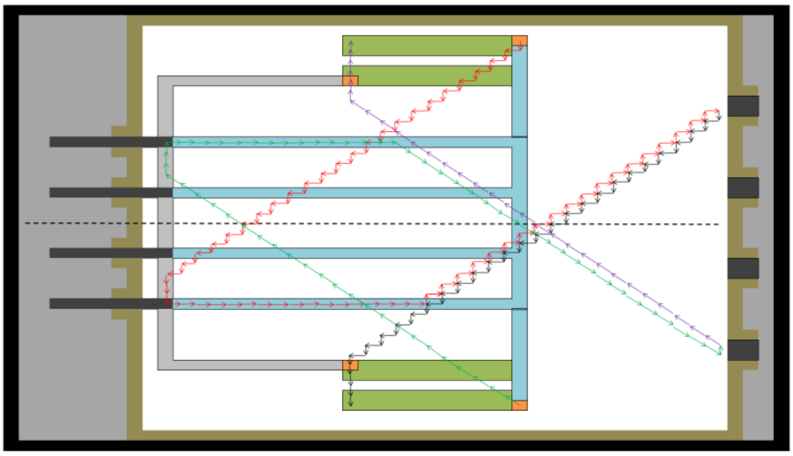

Figure C 2 - Stepwise and direct approach longest shortest paths

While traveling this path, the longest distance that a robot is loaded with a bag is 55 patches, which leaves 102 patches to be travelled by the robot without it having a load. Driving without a load consumes 0.00416667 % per seconds and a robot needs 102 seconds to travel a distance of 102 patches, meaning that the battery percentage decreases 102 * 0.00416667 = 0.42500034 percentage points. To travel one patch with a loaded bag reduces the battery percentage with 0.00555556 percentage points per second. The longest path contains 55 patches to be travelled while the robot is loaded, corresponding to 0.00555556 * 55 = 0.305558 percentage points per second. The total battery consumption of a robot that travels the longest possible path without visiting a charging area during this trip adds up to 0.73055614 percentage points per second. This means that when a robot is only traveling this longest path and keeps doing so without visiting a charging, it can drive 136 cycles on one charged battery for it to run fully empty. This corresponds to 136 * 157 = 21.352 seconds – or 355.9 minutes or almost 6 hours. An important note to this conclusion is that this assumes no idle time in between, so the robot is not in any queue or often needs to wait for other robots to pass by.

A threshold is set for the minimum battery percentage for a robot before it needs to return to one of the charging areas. When the battery percentage is lower than the battery percentage required for starting and completing the longest path, the robot is not allowed to accept a new transport request for an incoming bag. The battery percentage required to start and complete the longest path is calculated earlier and set to 0.73055614%. To allow for some slack and possible unforeseen circumstances like encountering a queue on the longest path, the value is increased by ten percent, setting the threshold at a rounded level of 0.8%.

Appendix D – Formalisation Examples

Two examples of formalisation of the agents and objects present in the system

Transport robot agents have:

- Size Integer =< patch size
- Speed Integer > 0 patches per tick (time step) = 1
- Battery power Integer >= 0 and =< variable integer representing full battery
- Availability Boolean, available or unavailable, determined by the combination of
    - Load Boolean, carrying one or no bag
    - Occupied Boolean, robot is already occupied with a transport task or not
    - Battery power Boolean, Integer >= minimum integer to accept a new bag assignment
- Phase List reporting on which of the six phases a robot is in
- Number Integer >= 0 and =< variable integer representing a maximum number
- Location Integer numbers representing the coordinates of the agent in the system
- Directions List of possible directions (N, NE, E, SE, S, SW, W, NW)

Bag objects have:

- Size Integer =< patch size
- Destination String, string of characters representing destination airport
- Directions List of possible directions, depending on its location
- Activity Boolean, a bag can be transported or be stationary at a time step
- Queuing time Integer >= 0 reporting how long a bag spends waiting on the belt
- Phase List reporting on where the bag is in the baggage handling process
- Location Integer numbers representing the coordinates of the object in the system
- Lead time Integer >= 0 reporting how much time a bag spends in the system
- Number Integer >= 0 and =< variable integer representing a maximum number

Appendix E – Pseudo Codes

Pseudo-code pick-up bags

```
; Initial setup
For all robots:
        set colour grey
        set phase "available"
        set current-path [ ]
        set loaded-bag 0
        set bag-limit 1
        set bag-transported " "
        set bag-assigned " "
        set makeup-station " "

For all bags
        set destination  one of the destinations A, B, C or D
        set my start belt one of the  drop off conveyor belts
        set phase "approaching belt"
        set robot-claimed " "

Start picking up a bag
        For each bag [
                if a bag is created
                        move to one of the conveyor belts
                if a bag reached one of the conveyor belts
                        set phase "on belt"
                if a bag arrives at the end of one of the drop-off belts
                        set phase "end of drop off belt no robot assigned"
                        send a signal into the system
                        ask the available robots to calculate their distance to the requesting bag
                        ask the robot with the smallest distance to the bag to pick me up ]
        For the nearest robot: pick me up [
                set  phase to "incoming request"
                set the requesting bag's phase to "end of drop off belt robot assigned"
                copy the id of the requesting bag and set it as bag-assigned to the robot
                set the  direction patch equal to the location of the requesting bag
                set the current path equal to the shortest path to the bag
                move over this shortest path

                if a robot reaches the bag
                        set phase "arrived at belt"
                        set loaded-bag 1
                        set the phase of the bag to "on robot"
                        copy the id of requesting bag and set it as bag-transported of the  robot]
End picking up a bag
```

Pseudo-code transporting and unloading bags

```
Start transporting and unloading a bag
    For robots with phase is "arrived at belt" and loaded-bag > 0 [
        Set makeup-station equal to destination of bag-here with a destination ]
    For robots with phase is "arrived at belt" [
        If the makeup-station is A and loaded-bag > 0
            Set direction-patch as entrance patches of makeup-station A
        If the makeup-station is B and loaded-bag > 0
            Set direction-patch as entrance patches of makeup-station B
        If the makeup-station is C and loaded-bag > 0
            Set direction-patch as entrance patches of makeup-station C
        If the makeup-station is D and loaded-bag > 0
            Set direction-patch as entrance patches of makeup-station D
        Set colour red
        Set phase to "transporting bag"
        Set current-path to find-a-path and set patch-here equal to my direction-patch
        Set the path of the bag with phase "on robot" equal to the robot's path ]
    For bags with phase "on robot" [
        Let path-for-the-bag be equal to the current-path of the robot that transports me
        Set current-path to path-for-the-bag ]
    For robots with patch-here is one of the entrance patches of either makeup-station [
        ask bag that has the same patch-here as me to die
        set loaded-bag equal to 0
        set colour grey
        set bag-transported " "
        set bag-assigned " "
        set phase "delivered bag"
        set makeup-station " "
        set direction-patch equal to one of the CS-areas patches that  aren't occupied yet
        set current-path find-a-path and set patch-here equal to my direction-patch ]
End transporting a bag
```

Pseudo-code updating battery level

```
; Initial setup
For all robots: set battery-level 100

Start updating battery level
For each robot [
        if "robot is available to accept a transport request and is located at one of the charging
        areas"
                increase the battery level with the chosen value for charging rate
        if "robot has accepted a request and is on its way to an incoming bag"
                decrease the battery level with the chosen value for battery reduction while
                driving empty
        if "robot arrived at the incoming conveyor belt and waits for a bag"
                decrease the battery level with the chosen value for battery reduction while
                waiting
        if "robot is driving to one of the make-up stations while carrying a bag"
                decrease the battery level with the chosen value for battery reduction while
                driving loaded
        if "robot delivered a bag and is driving back to one of the charging and storage areas"
                decrease the battery level with the chosen value for battery reduction while
                driving empty ]
End updating battery level
```

The values for the battery charging and reduction rates as argued in this section are linked to the phases in which a robot can be. The combination of possible robot phases and corresponding charging rates are summarized below.

Robot phases	**Charging rate**
Available	+ 0.00694444 % / sec
Incoming-request	- 0.00416667 % /sec
Arrived at belt	- 0.00138889 % / sec
Transporting bag	- 0.00555556 % / sec
Delivered bag	- 0.00416667 % /sec

Appendix F – Arrival Pattern

Table F 1 - Arrival Pattern Used

Nr.	Flight	Destination	Departure time	First check-in time	Last check-in time	AC type	Max Passengers
1	TRA6301	Gran Canaria	07:10	5:10	6:30	B737	189
2	TRA6771	Budapest	07:30	5:30	6:50	B737	189
3	TRA2N	Eindhoven	07:35	5:35	6:55	B737	189
4	CFE4452	London City	09:15	7:15	8:35	E190	98
5	CFE4454	London City	10:55	8:55	10:15	E190	98
6	TRA5605	Salzburg	13:05	11:05	12:25	B737	189
7	CFE4476	London City	13:50	11:50	13:10	E190	98
8	TRA6923	Innsbruck	16:40	14:40	16:00	B738	189
9	TRA5053	Alicante	17:10	15:10	16:30	B737	189
10	CFE4478	London City	17:40	15:40	17:00	E190	98
11	TRA5293	Vienna	18:25	16:25	17:45	B737	189
12	TRA6493	Venice	18:40	16:40	18:00	B737	189
13	CFE4458	London City	19:05	17:05	18:25	E190	98

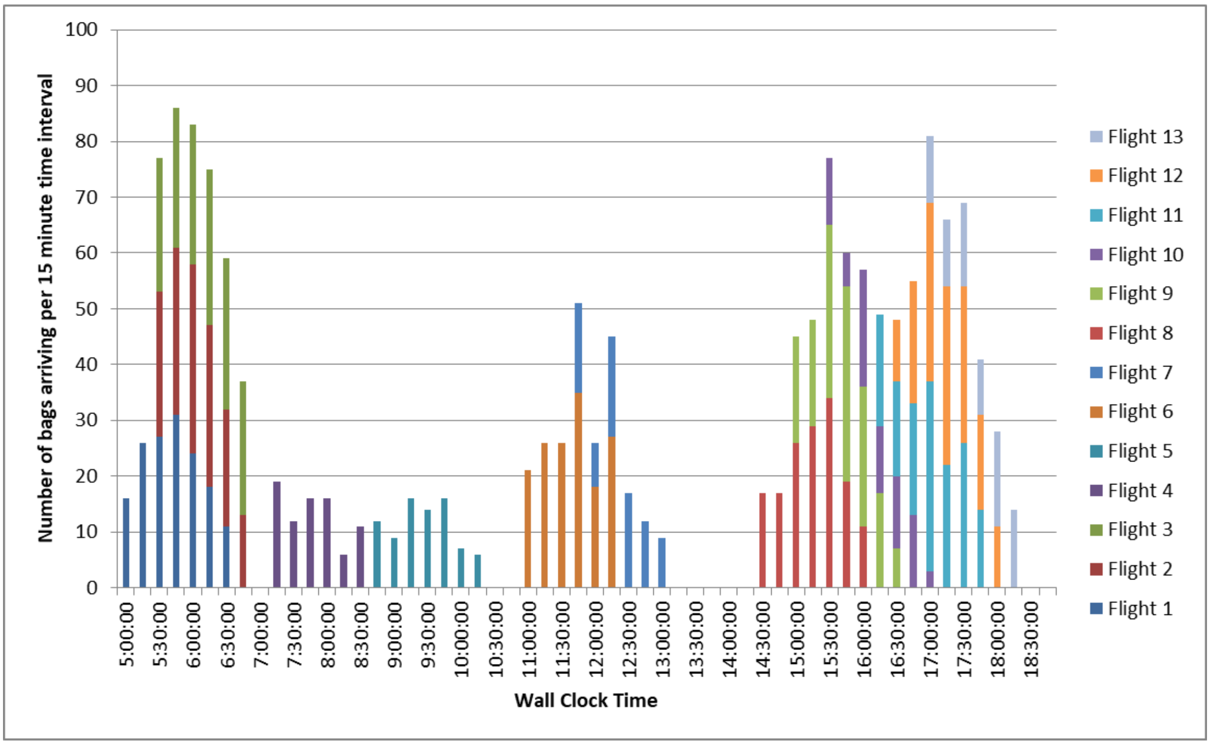

Figure F 1 - Visualisation of Arrival Pattern

Appendix G – Extreme Value for the Arrival Pattern

The same flights are considered, only the capacity of these flights are altered to represent an extreme situation in which all 13 flights use a Boeing 777-300 with 330 seats, all these seats are occupied and on average everyone on board takes one bag with them. This results in a total of 4290 seats, or 4290 bags to be distributed over the 13 flights. Compared to the realistic flight, passenger and bag distribution used in the first experimental design, where 1624 bags were distributed over the 13 flights, the number of bags to be handled increased by 164%.

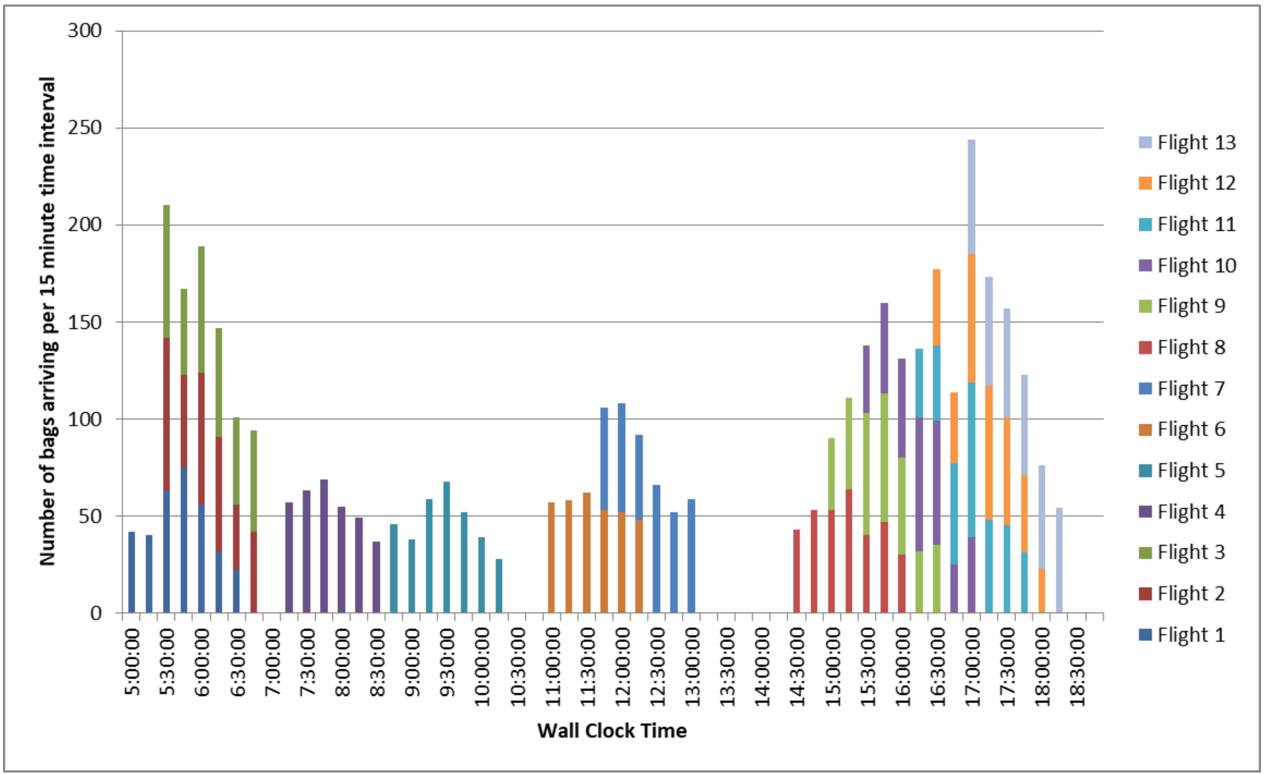

Figure G 1 - Visualisation of Extreme Arrival Pattern

Appendix H – Experimental Design Parameter Sweep Experiment

Nr.	# Bags	Configuration	# Robots	Battery Reduction Idle [%/sec]	Battery Reduction Empty [%/sec]	Battery Reduction Loaded [%/sec]	Charging Rate [%/sec]	Battery Level Threshold [%]
1	4290	Four Horizontal Areas	34	0.00119	0.00550	0.00258	0.00339	49
2	4290	Six Vertical Areas	86	0.00192	0.00617	0.00551	0.00944	21.9
3	4290	Four Horizontal Areas	11	0.00908	0.00735	0.00736	0.00486	97.6
4	1624	Six Horizontal Areas	53	0.00273	0.00847	0.00465	0.00736	77.7
5	1624	Four Horizontal Areas	63	0.00852	0.00998	0.00892	0.00830	60.8
6	1624	Six Vertical Areas	21	0.00154	0.00901	0.00430	0.00094	84.2
7	1624	Six Vertical Areas	27	0.00495	0.00662	0.00105	0.00217	93.2
8	1624	Four Horizontal Areas	30	0.00032	0.00147	0.00066	0.00729	56.7
9	4290	Central Areas	8	0.00789	0.00422	0.00928	0.00622	10.2
10	4290	Central Areas	42	0.00440	0.00398	0.00639	0.00408	89.3
11	4290	Four Horizontal Areas	78	0.00890	0.00880	0.00843	0.00198	27.6
12	1624	Central Areas	19	0.00636	0.00955	0.00341	0.00669	38.3
13	4290	Four Horizontal Areas	46	0.00576	0.00719	0.00685	0.00330	7.7
14	1624	Six Vertical Areas	66	0.00679	0.00577	0.00318	0.00799	32.6
15	4290	Six Vertical Areas	57	0.00762	0.00075	0.00211	0.00066	70
16	1624	Four Horizontal Areas	73	0.00620	0.00777	0.00299	0.00653	44.6
17	1624	Central Areas	76	0.00048	0.00028	0.00030	0.00838	52.3
18	1624	Six Horizontal Areas	82	0.00509	0.00207	0.00142	0.00291	54
19	1624	Six Horizontal Areas	43	0.00210	0.00064	0.00492	0.00123	64.9
20	4290	Four Horizontal Areas	91	0.00338	0.00274	0.00523	0.00547	34.8
21	1624	Four Horizontal Areas	4	0.00941	0.00700	0.00368	0.00022	25.9
22	1624	Six Horizontal Areas	15	0.00718	0.00239	0.00780	0.00514	17.6
23	4290	Four Horizontal Areas	2	0.00565	0.00188	0.00631	0.00577	5
24	4290	Six Vertical Areas	59	0.00400	0.00807	0.00810	0.00260	81.6
25	1624	Six Vertical Areas	93	0.00073	0.00304	0.00182	0.00143	93.4
26	1624	Six Horizontal Areas	69	0.00807	0.00453	0.00726	0.00444	3
27	4290	Six Vertical Areas	89	0.00379	0.00495	0.00568	0.00897	66.7
28	4290	Six Vertical Areas	51	0.00970	0.00127	0.00089	0.00921	16.3
29	4290	Central Areas	37	0.00251	0.00344	0.00977	0.00973	73.9
30	4290	Six Vertical Areas	26	0.00328	0.00528	0.00940	0.00391	40.2

Figure H 1 - Experimental Design Parameter Values

Appendix I – Model Verification Tests

Verification tests

1. Recording and Tracking Agent Behaviour
2. Single-Agent Testing
3. Interaction Testing in a Minimal Model
4. Multi-agent Testing

1. Recording and Tracking Agent Behaviour

To check if the model on agent level is working as expected the behaviour of individual agents can be investigated. To verify the model on the agent level, the inputs, states and outputs of the agents – in this research the robots – are recorded and logged. This can be done by recording the behaviour of the robot 'from the outside' by following individual robots around in the model using the 'watch agent' tool in NetLogo. Besides this, the internal processes of individual agents can be logged by using the 'inspect agent' tool in NetLogo. This tool shows the 'thought' processes that happen within an agent during runtime.

2. Single-Agent Testing

Most agent-based models consist of a large number of agents. In almost all studies that use agent-based models, the behaviour of all these agents combined is tested for verification of the model. However, testing the behaviour of a single agent is often overlooked, despite it being an important action in the verification of agent-based models. One way to test the single agent behaviour is by means of 'unit testing', this can be done continuously and automatically. Unit testing entails the addition of a test line in the code of the model that gives insight into how that specific part of the code responds to these tests. In NetLogo, the 'show' function can be used in the code, which will show the results of all the unit tests in the interface during runtime.

3. Interaction Testing in a Minimal Model

The same technique of unit testing can be applied to multiple robots. As the agent-based model developed only contains one agent type, the minimal model consists of two agents. This verification test checks whether or not the expected and modelled agent interactions happen correctly. Two checks in this type of verification test are if the model shows the desired interaction and whether or not the model shows undesired interactions.

4. Multi-agent Testing

Verification tests one to three focussed on single agents or the minimal model. The final verification test is to check the entire model on whether or not it does what it should do. To do so, input parameters can be varied and a theoretical prediction of the effect of these variations on the model behaviour and output can be formulated. If after running the model the behaviour and output correspond to the theoretical prediction, the agent-based model passed the multi-agent verification test successfully.

Table I 1 - Verification Tests and Results

Nr.	Test Type	Description	Expected Result	Obtained Result	Verified?	Resolved?
1	1	Every robot has a phase that changes depending on its actions. This test checks if robots go through all phases when they execute a bag transport task.	Robots that are assigned to a transport task go through the following phases in order: available, incoming request, arrived at belt, transporting bag, delivered bag, en route to charging and storage area, charging.	This test is executed multiple times for different robots. All robots tested successfully go through the six phases in that sequence	Yes	n/a
2	1	Every robot can carry one bag at a time and has an attribute 'loaded bag' which can only have the values 0 or 1. This test checks if the 'loaded bag' attribute is measured and updated correctly throughout a run.	Robots initially have a value of 0 for the attribute 'loaded bag'. If a robot arrived at a belt and picks up a bag this value is changed to 1 and when it drops the bag at one of the makeup stations the value changes bag to 0.	This test is executed multiple times for different robots. All robots successfully changed their loaded bag attribute from 0 to 1 and back to 0 when transporting a bag. For none of the tested robots the loaded bag attribute got a value greater than 1 at any point.	Yes	n/a
3	1	Every robot has a battery level which decreases when not charging and increases when charging. This test checks if the value of the battery level attribute is measured and updated correctly throughout a run.	Robots initially have a value of 100 for the attribute 'battery level'. When moving the battery level decreases at the rate specified by the slider for the corresponding movement type (standing idle, moving without a bag, moving with a bag) in the interface view. When the value is below 100 and a robot is at a storage and charging position, the battery level increases with the rate specified by the 'charging rate' slider.	This test is executed multiple times for different robots. The battery level initially is 100 and decreases according to the set rates. However, when it is at a charging and storage area it can exceed a value of 100 for the battery level as the charging phase is only connected to the charging and storage area positions and does not check if the robot actually needs charging. This is resolved by checking if the robot actually needs charging (when its battery level < 100) before starting to charge.	No	Yes
4	1	Like robots, every bag can have	Bags start counting the time they	Before bags are picked up the	Yes	n/a

		different phases. One of the phases is 'on robot' which measures the time a robot spend on a bag so in the moving sorting process. In this test it is checked if the counter that counts the time steps a bag spends on a robot starts at the moment the bag is picked up by a robot, and if it's updated every tick and stops updating when the bag is no longer on a robot.	spend on a robot the moment they are loaded onto a robot. While being transported by the robot, the 'on robot time' attribute increases with 1 every time step. When the robot drops the bag at one of the chutes to the makeup stations, the counter stops counting and saves the value for 'on robot time' so it can be used later to calculate the average 'on robot time' of all the robots in the system.	value for their attribute 'on robot time' is 0. Whenever their status changes to 'on robot' when they are picked up by a robot, the counter starts adding 1 to the 'on robot time' attribute. When the bag reaches one of the chutes to the makeup station, the bag dies (is deleted from the model) and the counting immediately stops. The end value is stored and used to calculate and update the average 'on robot time' of all robots in the system.		
5	1	Every robot has a priority attribute which can be either 0 or 1. When a robot is not transporting a bag, the priority is 0 and when a robot is transporting a bag, the value for the priority attribute is 1.	The initial value for the robot's attribute 'priority' is 0 and remains 0 until it picks up a bag and changes its phase to 'transporting bag', that's when the priority attribute changes to 1. When the robot's phase is changed to 'delivered bag' as it drops off a bag at a makeup station, the value for priority becomes 0 again.	This test is executed multiple times for different robots. When following a robot around by using the 'inspect agent' function in NetLogo, the value for the priority attribute changes correctly from 0 to 1 when it picks up a bag and back to 0 when it drops off a bag.	Yes	n/a
6	2	When a robot picks up a bag it should translate the destination of the bag into the corresponding makeup station. This test checks if individual robots translate bag destination A to makeup station A, and so on.	When a robot picks up a bag at the end of one of the incoming conveyor belts, it copies the 'destination' attribute of the bag and translates it to a robot attribute 'makeup-station'. Destination A is translated to makeup station A, B to B, C to C and D to D.	This test is executed multiple times for different robots. When a robot reaches a bag at one of the incoming conveyor belts, its status changes from 'incoming request' to 'arrived at belt' and 'transporting bag' and the attribute 'makeup-station' is filled with A, B, C or D, depending on the value for the 'destination' attribute of the bag it picks up.	Yes	n/a

7	2	Only bags that are available to accept a transport request are allowed to be called to pick up an incoming bag. This test checks if only available robots are called by showing a list of available robots (so robots with phase "available" and a battery level that is equal to or exceeds the set battery level threshold) at every tick.	Whenever a bag reaches the end of an incoming conveyor belt a list of available robots is displayed in the command centre of the interface. Only robots from this list can claim a bag. When a robot claimed a bag it's no longer available so in the next tick that robot is eliminated from the list with available robots.	This test is executed multiple times for multiple robots. When a robot accepts a transport request its phase changes to 'incoming request', eliminating this robot from the list with 'available robots'.	Yes	n/a
8	2	When a robot picks up a bag, the bag is placed on the robot and is transported to one of the makeup stations. Once the robot with the bag arrives at one of these makeup stations, the bag needs to be unloaded and visually disappear from the model. This test checks if the bag visually is placed on a robot and unloaded once a makeup station is reached.	It is expected that when a robot picks up a bag, the visual of the (red) bag disappears and the colour of the robot changes from grey to red. When the bag is dropped off at one of the makeup stations, the colour of the robot should turn from red back to grey.	This test is executed multiple times for multiple robots. In each test the colour of the robot changes from grey to red when it picks up a bag, and from red back to grey when the bag is dropped at one of the makeup stations.	Yes	n/a
9	2	When a bag reaches the end of an incoming conveyor belt, it should stop moving and wait at that last patch to be picked up by a robot. Once the bag is picked up, the next bag in line can move to the last patch of the conveyor belt that was vacated when the bag in front was picked up by a robot. This test checks if the desired bag behaviour is executed by the bags.	Upon reaching the end of a conveyor belt, bags should stop moving. Only one bag can be at the final position of each conveyor belt. Once it's picked up by a robot that final position is vacated and can be occupied by the next bag in line.	Upon reaching the end of an incoming conveyor belt, bags don't stop moving but instead keep moving forward and enter the baggage handling area on their own. This problem is resolved by marking the first patch after the last conveyor belt patches 'true' for the patch attribute 'occupied for bags?'. By doing so, bags that are located at the end of an incoming conveyor belt are restricted to move passed the last patch of the conveyor belt, making them stand idle at the last	No	Yes

				patch.		
10	2	For robots to store the shortest path they calculated, a robot attribute called 'current-path' is filled with the coordinates of the patches in its path. This list of coordinate shows the sequence in which these patches have to be taken by the robot. This test checks if the 'current-path' attribute is updates when a new shortest path is calculated so the robot knows where to go next.	The attribute 'current-path' should initially be empty and filled based on the destination patch of the robot. Once a destination patch is chosen, the robot calculates its shortest path to that destination patch and this shortest path is stored in the 'current-path' attribute. By calling the show function of NetLogo the current path lists of all robots can be inspected which can't be empty.	When a robot can't find a destination patch, the 'current-path' attribute list is emptied, seen from the show function in NetLogo. This attribute being empty causes robots to stand still with no idea where to go next, forming an obstacle to other robots as it doesn't move as it doesn't know where to go. This has been resolved by adding a 'temporary direction patch' attribute to the robots that is filled whenever the initial desired destination patch is not available. This way, the 'current-path' attribute is always filled so the robot keeps moving towards its goal.	No	Yes
11	3	Not more than one robot is allowed to be on one patch. This test checks if the patch attribute 'occupied-for-robots?' is true whenever a robot is on that specific patch. If a patch is occupied by a robot, another robot cannot move to that same patch while it is still occupied by the first robot.	The attribute 'occupied-for-robots?' of the patch a robot stands on is true if a robot has the coordinates of that patch. A robot cannot move to or move over a patch that is true for the 'occupied-for-robots?' attribute.	Robots don't move to or over patches that are occupied by other robots, except at storage and charging areas. Multiple robots have the same coordinates, meaning that a patch is occupied by more than one robot. This is resolved by restricting a robot to pick an occupied patch as destination patch in the charging and storage areas	No	Yes
12	3	Every robot has a priority (1 in case it's transporting a bag, 0 otherwise). In case two robots want to go to the same patch	Robots with a value of 0 for the priority attribute have to give way to a robot with a priority value of 1 in case of an imminent collision.	This test is executed multiple times for different situation. In one of the situations the following situation occurred:	Yes	n/a

		simultaneously, the collision avoidance algorithm is evoked, in which the robot that is carrying a bag gets priority over the other robot, which has to wait until the robot with the priority passed.		Robot 6 (not carrying a bag) and robot 7 (carrying a bag) both want to go to the same patch, patch 44 12. In the next tick robot 7 moved to patch 44 12 while robot 6 had to wait. The same pattern was found for the other situations and other robots this test was executed for.		
13	3	This test checks if a collision avoidance measure is evoked when the trajectories of two robots overlap and robots are expected to reach one of these overlapping patches at the same time.	When two robots have overlapping trajectories and are expected to end up at one of these overlapping patches at the same time, one of the robots have to alter its trajectory, avoiding a possible collision.	This test is executed multiple times for different situation. The trajectory, stored in the 'current-path' attribute of the robots is altered when parts of the trajectories of two robots overlapped at the same time.	Yes	n/a
14	4	All robots calculate their shortest path between destinations and have to stick to that path. This test checks if all robots construct and follow their shortest path by colouring the intended shortest paths and visually checking if no robot diverts from their coloured path.	All robots that are assigned to a transport task calculate a shortest path to the bag and take this path to pick up the bag, after which it again calculates a shortest path to the appropriate makeup station, takes it, and calculates a shortest path back to a charging and storage position. No robots should be visible outside the coloured paths.	No robots are visible on white patches (white marks the patches that are not used for any shortest path). This means that all robots follow their shortest path. No unexpected diversions are detected.	Yes	n/a
15	4	Every robot should calculate and travel over a shortest path from one to another destination. Destinations can be a charging and storage position, an incoming conveyor belt or a makeup station. In all four of the layouts, robots have therefore no reason	When running the model, no robots are expected to be at or travel to one of the outer corners of the baggage handling area. They should only move between the incoming conveyor belts, makeup stations and charging and storage positions.	This test is executed multiple times. In none of the runs executed robots end up or travel to one of the outer corners or to another location in the baggage handling area where they shouldn't be.	Yes	n/a

		to pick another destination within the baggage handling area so. This test checks if robots do not travel to or through one of the outer corners of the baggage handling area, marked with white patches.				
16	4	The collision avoidance measures are focussed on collisions between two robots. This checks if no deadlock situations occur when more than two robots run through a collision avoidance measure.	When more than two robots are in each other's proximity, collision avoidance measures are invoked. It is expected that robots do not end up in a deadlock but in a near collision situation where they have to get out one by one by looping through the collision avoidance measures.	This test is executed multiple times under varying input parameter settings. With a strict right moving policy for frontal collisions, deadlocks could occur. This is resolved by relaxing the collision avoidance measure of frontal collisions to a move at any empty surrounding patch, instead of forcing the robots to move to the right. That patch on the right could be occupied at that time, which caused deadlock situations.	No	Yes

In this example the minimum value for the KPI 'average process time of bags' is calculated for the four layout configurations. The time it takes for bags to get to the belt, to move over the belt, to wait for robot assignment and the minimum time bags spend on a robot are equal for all layout configurations and calculated by counting the number of patches needed to travel this distance. The minimum average process time of bags can be calculated by:

$$min.\,time\ to\ belt + min.\,time\ on\ belt + min.\,time\ waiting\ for\ assignment + min.\,time\ waiting\ on\ robot + min.\,time\ on\ robot$$
$$4 + 7 + 1 + min.\,time\ waiting\ on\ robot + 39$$

Table I 2 - Manual KPI Calculation Example

	Min. time waiting on robot	**Total minimum average process time of bags**
Central Areas	17	4 + 7 + 1 + 17 + 39 = **68 seconds**
Four Horizontal Areas	4	4 + 7 + 1 + 4 + 39 = **55 seconds**
Six Horizontal Areas	3	4 + 7 + 1 + 3 + 39 = **54 seconds**
Six Vertical Areas	4	4 + 7 + 1 + 4 + 39 = **55 seconds**

Appendix J – Model Validation Questionnaire

<u>The Baggage Robot Concept</u>

Baggage Handling Systems

1. What is your expertise in the field of conventional baggage handling systems?
2. What are in your opinion the main advantages of using conveyor belts in baggage handling systems?
3. What are in your opinion the main disadvantages conventional baggage handling systems?

Individual Transport Robots

4. What is your experience or expertise in systems with individual transport robots?
5. Do you think individual transport robots can be useful in baggage handling systems? If so, where in the process of handling baggage can individual transport robots be of most value?

The Baggage Robot Concept

6. Are the formulated KPIs adequate to judge the performance of the baggage robot concept?

<u>Simulation model of the sorting process in the Baggage Robot concept</u>

7. Which input parameters are of most importance for a simulation model to investigate the use and performance of autonomous and individual transport robots in baggage handling systems?
8. Which key performance indicators are important to be generated from the simulation model?
9. To what extent does the simulation model represent the sorting process of a baggage handling system that makes use of autonomous and individual transport robots to replace conveyor belts?
10. Can this simulation model of the sorting process of the baggage robot concept proof that it is an adequate substitution for conveyor belts in baggage handling systems?
11. What experiments do you think should be executed in order to conclude whether or not the baggage robot concept is an adequate substitute for a conventional baggage handling system?
12. Any other remarks/considerations you would like to make.

Appendix K – Experimental Results

APT = Average Process Time of Bags
AMH = Average Percentage of Mishandled Bags

ALT = Average Percentage of Loaded Trips
AET = Average Percentage of Empty Trips
ACT = Average Percentage of Charging Time

K.1. Infeasible Combinations for 1624 and 4290 bags in the Parameter Sweep Experiment

Table K 1 - Infeasible Combinations for both scenarios in the Parameter Sweep Experiment

Configuration	# of bags	# of robots	Battery reduction idle [%/sec]	Battery reduction empty [%/sec]	Battery reduction loaded [%/sec]	Charging rate [%/sec]	Battery level threshold [%]	Bag Statistics		Robot Statistics		Number of bags handled in runtime	
Four Horizontal Areas	1624	4	0.00941	0.007	0.00368	0.00022	25.9	APT [sec]	4338.0	ALT [%]	42.1%	Min	552
												Max	454
										AET [%]	57.9%	Average	558.2
								AMH [%]	75,96%			Median	558.5
										ACT [%]	69.9%	Stdev.	2.7
												Var.	7.1
Six Vertical Areas	1624	21	0.00154	0.00901	0.0043	0.00094	84.2	APT [sec]	248.6	ALT [%]	44.5%	Min	1569
												Max	1602
										AET [%]	55.5%	Average	1585.8
								AMH [%]	9,94%			Median	1587.0
										ACT [%]	76.9%	Stdev.	7.8
												Var.	61.6
Four Horizontal Areas	4290	2	0.00565	0.00188	0.00631	0.00577	5	APT [sec]	12266.1	ALT [%]	41.1%	Min	747
												Max	758
										AET [%]	58.9%	Average	753.6
								AMH [%]	85,48%			Median	754
										ACT [%]	18.8%	Stdev.	2.2
												Var.	4.9
Central Areas	4290	8	0.00789	0.00422	0.00928	0.00622	10.2	APT [sec]	6358.5	ALT [%]	39.8%	Min	2295
												Max	2313
										AET [%]	60.2%	Average	2304
								AMH [%]	72,80%			Median	2303.5
										ACT [%]	35.2%	Stdev.	3.9
												Var.	15.2
Four Horizontal Areas	4290	11	0.00908	0.00735	0.00736	0.00486	97.6	APT [sec]	9757.3	ALT [%]	41.9%	Min	2079
												Max	2102
										AET [%]	58.1%	Average	2088.3
								AMH [%]	73,89%			Median	2088.0
										ACT [%]	59.0%	Stdev.	4.6
												Var.	20.8

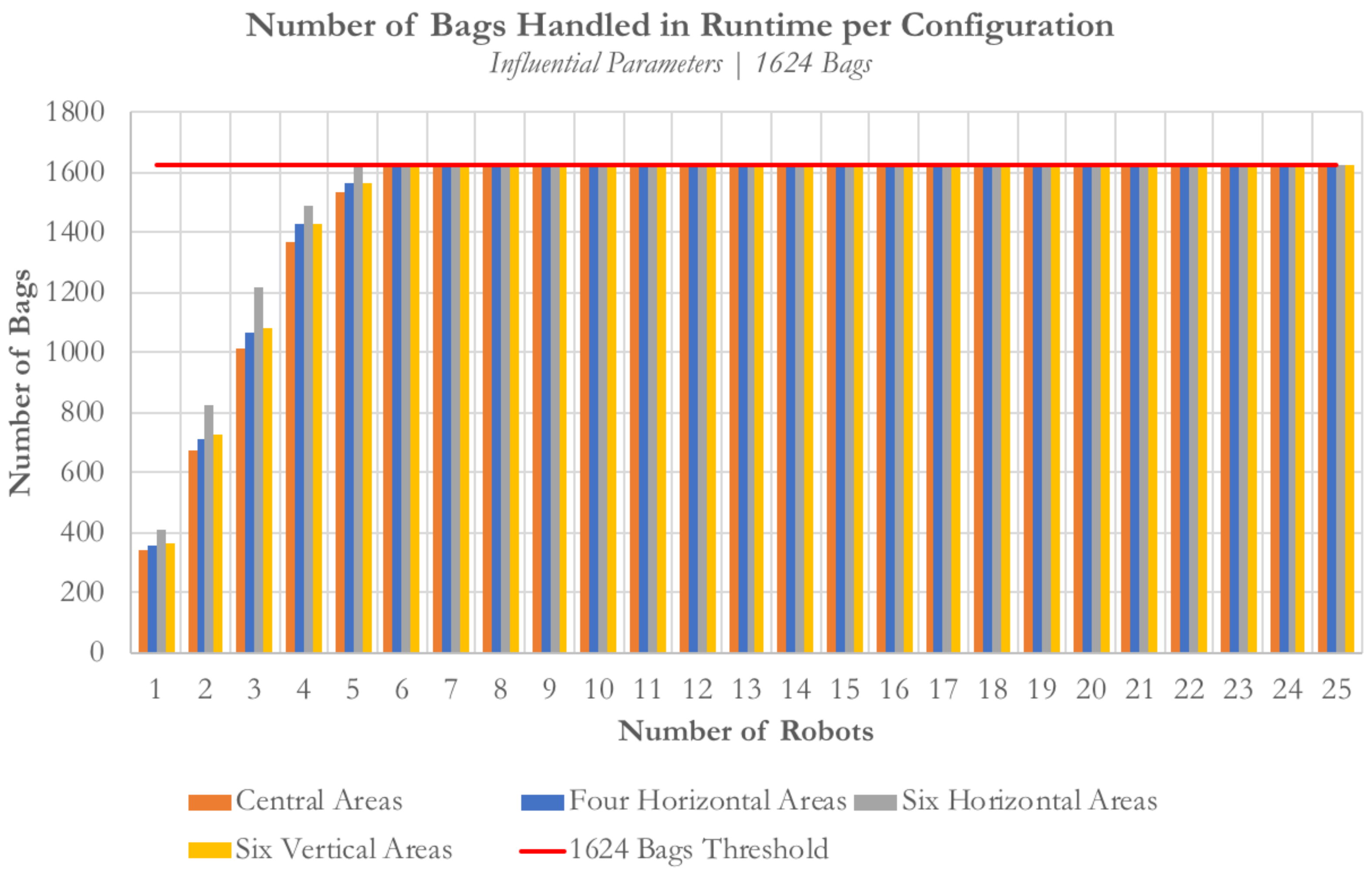

Figure K 1 - Number of Bags Handled in Runtime per Layout Configuration for the 1624 Bags Scenario in Experiment Design III

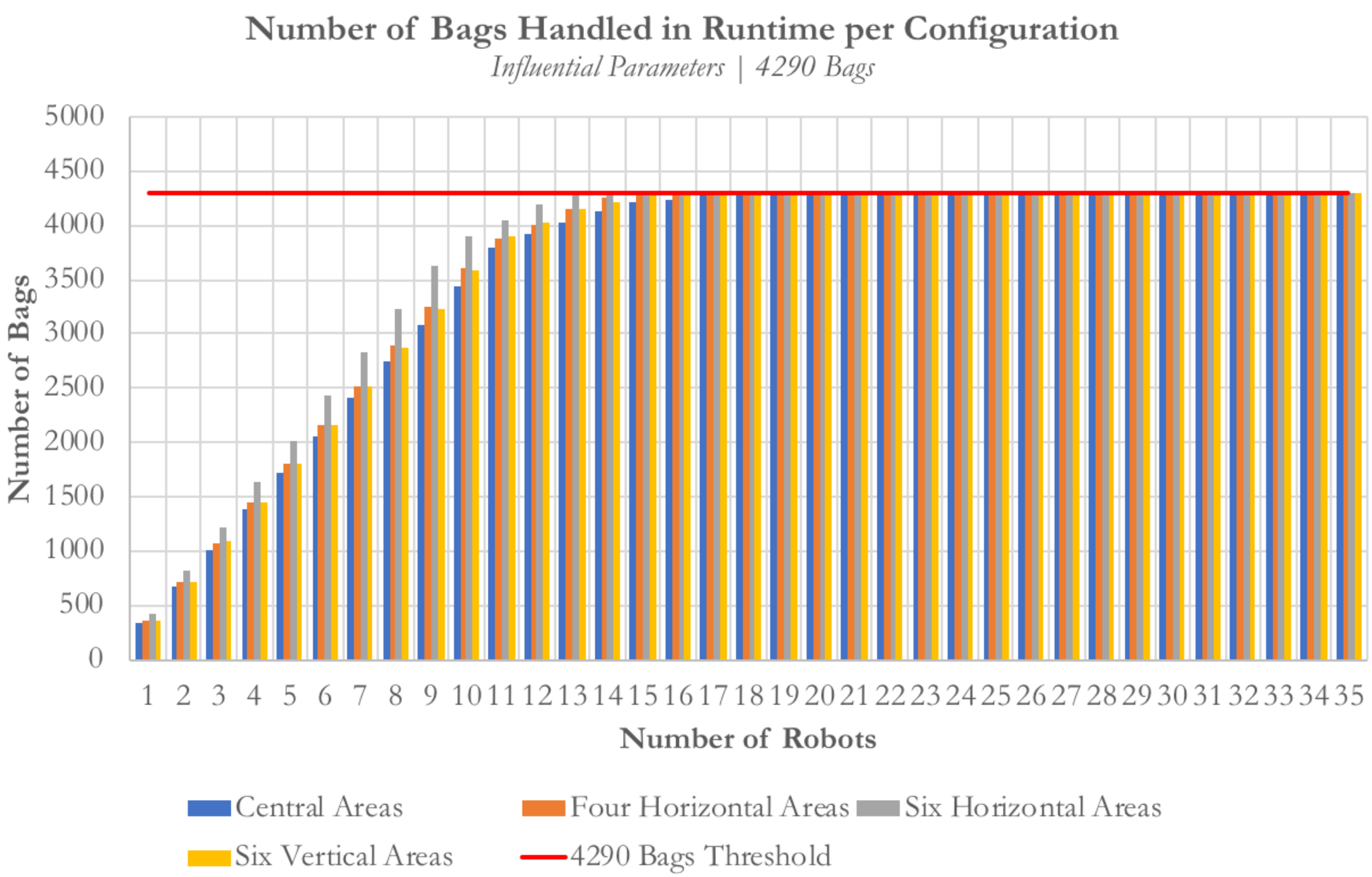

Figure K 1 - Number of Bags Handled in Runtime per Layout Configuration for the 4290 Bags Scenario in Experiment Design III

K.2. Combinations of Input Parameter Values Causing Optimal Values for 1624 Bags

Please note that only the feasible combinations of input parameters are shown.

Figure K 3 - Feasible Combinations of Input Parameter Values Causing Optimal Values for the 1624 Bags Scenario

1624 Bags			Input Parameter Values							Values for the other KPIs on this combination of input parameter values						
KPIs	**Theoretical Optimal Value**	**Optimal Value**	**# of Robots**	**Configuration**	**Battery reduction idle**	**Battery reduction empty [%/sec]**	**Battery reduction loaded [%/sec]**	**Charging rate [%/sec]**	**Battery level threshold**	**Average Process Time of Bags**	**# of Robots**	**% Empty Trips**	**% Loaded Trips**	**% Charging Time**	**% Mishandled Bags**	**Number of Avoided Conflicts**
Average Process Time of Bags	≤ 6 minutes or 3600 seconds	61.75 sec	53	Six Horizontal Areas	0.00273	0.00847	0.00465	0.00736	77.7	x	53	51.57%	48.43%	5.03%	0.00%	1643
Number of Robots	As small as possible	15	15	Six Horizontal Areas	0.00718	0.00239	0.0078	0.00514	17.6	65.05 sec	x	51.15%	48.85%	19.27%	0.00%	1406
% Empty Trips	Close to 50%	50.94%	15	Six Horizontal Areas	0.00718	0.00239	0.0078	0.00514	17.6	65.25 sec	15	x	49.06%	19.36%	0.00%	1345
% Loaded Trips	As high as possible	49.06%	15	Six Horizontal Areas	0.00718	0.00239	0.0078	0.00514	17.6	65.25 sec	15	50.94%	x	19.36%	0.00%	1345
% Charging Time	As low as possible	0.14%	76	Central Areas	0.00048	0.00028	0.0003	0.00838	52.3	91.10 sec	76	57.28%	42.72%	x	0.00%	1267
% Mishandled Bags	Between 0 and 0.02%	0.00%	*All combinations of input parameter values score 0.00% on the KPI Percentage of Mishandled Bags*													
Number of Avoided Conflicts	As low as possible	1126	19	Central Areas	0.00636	0.00955	0.00341	0.00669	38.3	92.88 sec	19	57.63%	42.37%	18.51%	0.00%	x

While complying with the optimal values of the other KPIs:

Table K 2 - Combinations of Input Parameter Values to Comply with All KPI Thresholds for the 1624 Bags Scenario

1624 Bags			**Input Parameter Values**						
KPI	**Theoretic Optimal Value**	**Optimal Value**	**# of Robots**	**Configuration**	**Battery reduction idle [%/sec]**	**Battery reduction empty [%/sec]**	**Battery reduction loaded [%/sec]**	**Charging rate [%/sec]**	**Battery level threshold [%]**
Average Process Time of Bags	≤ 6 minutes or 3600 seconds	61.7 sec	53	Six Horizontal Areas	0.00273	0.00847	0.00465	0.00736	77.7
Number of Robots	As small as possible	15	15	Six Horizontal Areas	0.00718	0.00239	0.0078	0.00514	17.6
% Empty Trips	Close to 50%	50.9%	15	Six Horizontal Areas	0.00718	0.00239	0.0078	0.00514	17.6
% Loaded Trips	As high as possible	49.1%	15	Six Horizontal Areas	0.00718	0.00239	0.0078	0.00514	17.6
% Charging Time	As low as possible	0.14%	76	Central Areas	0.00048	0.00028	0.0003	0.00838	52.3
Number of Avoided Conflicts	As low as possible	1126	19	Central Areas	0.00636	0.00955	0.00341	0.00669	38.3
% Mishandled Bags	Between 0 and 0.02%	0.00%	All combinations of input parameter values score 0.00% on the KPI Percentage of Mishandled Bags						

K.3. Combinations of Input Parameter Values Causing Optimal Values for 4290 Bags

Please note that only the feasible combinations of input parameters are shown.

Figure K 4 - Feasible Combinations of Input Parameter Values Causing Optimal Values for the 4290 Bags Scenario

4290 Bags			Input Parameter Values							Values for the other KPIs on this combination of input parameter values						
KPIs	Theoretical Optimal Value	Optimal Value	# of Robots	Configuration	Battery reduction idle	Battery reduction empty [%/sec]	Battery reduction loaded [%/sec]	Charging rate [%/sec]	Battery level threshold	Average Process Time of Bags	# of Robots	% Empty Trips	% Loaded Trips	% Charging Time	Number of Avoided Conflicts	% Mishandled Bags
Average Process Time of Bags	≤ 6 minutes or 3600 seconds	77.96 sec	86	Six Vertical Areas	0.00192	0.00617	0.00551	0.00944	21.9		86	55.15%	44.85%	6.14%	8765	0.00%
Number of Robots	As small as possible	26	26	Six Vertical Areas	0.00328	0.00528	0.0094	0.00391	40.2	90.26 sec		56.00%	44.00%	46.32%	8251	0.00%
% Empty Trips	Close to 50%	54.96%	57	Six Vertical Areas	0.00762	0.00075	0.00211	6.60E-04	70	79.60 sec	57		45.04%	27.92%	8806	0.00%
% Loaded Trips	As high as possible	45.04%	57	Six Vertical Areas	0.00762	0.00075	0.00211	6.60E-04	70	79.60 sec	57	54.96%		27.92%	8806	0.00%
% Charging Time	As low as possible	1.93%	51	Six Vertical Areas	0.0097	0.00127	0.00089	0.00921	16.3	79.55 sec	51	55.09%	44.91%		8547	0.00%
Number of Avoided Conflicts	As low as possible	5042	37	Central Areas	0.00251	0.00344	0.00977	0.00973	73.9	323.35 sec	37	57.45%	42.55%	15.16%		12.21%
		7733	34	Four Horizontal Areas	0.00119	0.0055	0.00258	0.00339	49	82.05 sec	34	55.30%	44.70%	30.77%		0.00%
% Mishandled Bags	Between 0 and 0.02%	0.02%	26	Six Vertical Areas	0.00328	0.00528	0.0094	0.00391	40.2	89.86 sec	26	55.99%	44.01%	46.37%	8225	
		0.00%	26	Six Vertical Areas	0.00328	0.00528	0.0094	0.00391	40.2	90.29 sec	26	56.00%	44.00%	46.32%	8252	
			34	Four Horizontal Areas	0.00119	0.0055	0.00258	0.00339	49	82.04 sec	34	55.37%	44.63%	30.95%	8124	
			46	Four Horizontal Areas	0.00576	0.00719	0.00685	0.0033	7.7	80.53 sec	46	55.28%	44.72%	37.92%	8146	
			51	Six Vertical Areas	0.0097	0.00127	0.00089	0.00921	16.3	79.55 sec	51	55.09%	44.91%	1.94%	8537	
			57	Six Vertical Areas	0.00762	0.00075	0.00211	0.00066	70	79.17 sec	57	55.08%	44.92%	27.92%	8614	
			59	Six Vertical Areas	0.004	0.00807	0.0081	0.0026	81.6	79.99 sec	59	55.24%	44.76%	40.32%	8547	
			78	Four Horizontal Areas	0.0089	0.0088	0.00843	0.00198	27.6	81.04 sec	78	55.41%	44.59%	40.66%	8088	
			86	Six Vertical Areas	0.00192	0.00617	0.00551	0.00944	21.9	78.90 sec	86	55.15%	44.85%	6.16%	8783	
			89	Six Vertical Areas	0.00379	0.00495	0.00568	0.00897	66.7	78.81 sec	89	55.16%	44.84%	5.63%	8801	
			91	Four Horizontal Areas	0.00338	0.00274	0.00523	0.00547	34.8	79.99 sec	91	55.28%	44.72%	6.61%	8238	

While complying with the optimal values of the other KPIs:

Table K 3 - Combinations of Input Parameter Values to Comply with All KPI Thresholds for the 4290 Bags Scenario

4290 Bags			**Input Parameter Values**						
KPI	**Theoretic Optimal Value**	**Optimal Value**	**# of Robots**	**Configuration**	**Battery reduction idle [%/sec]**	**Battery reduction empty [%/sec]**	**Battery reduction loaded [%/sec]**	**Charging rate [%/sec]**	**Battery level threshold [%]**
Average Process Time of Bags	≤ 6 minutes or 3600 seconds	77.96 sec	86	Six Vertical Areas	0.00192	0.00617	0.00551	0.00944	21.9
Number of Robots	As small as possible	26	26	Six Vertical Areas	0.00328	0.00528	0.0094	0.00391	40.2
% Empty Trips	Close to 50%	55.96%	57	Six Vertical Areas	0.00762	0.00075	0.00211	0.00066	70
% Loaded Trips	As high as possible	54.04%	57	Six Vertical Areas	0.00762	0.00075	0.00211	0.00066	70
% Charging Time	As low as possible	1.93%	51	Six Vertical Areas	0.0097	0.00127	0.00089	0.00921	16.3
Number of Avoided Conflicts	As low as possible	7733	37	Central Areas	0.00251	0.00344	0.00977	0.00973	73.9
% Mishandled Bags	Between 0 and 0.02%	0.002%	26	Six Vertical Areas	0.00328	0.00528	0.0094	0.00391	40.2
		0.00%	Multiple (10) combinations of input parameter values score 0.00% on the KPI Percentage of Mishandled Bags. See Appendix K.4. for these 10 combinations.						

K.4. Combinations of Input Parameter Values Causing 0% Mishandled Bags for 4290 Bags

Table K 4 - Input Parameter Value Combinations Causing a Value of 0.00% for the KPI Mishandled Bags for the 4290 Bags Scenario

4290 Bags			Input Parameter Values						
KPI	**Theoretic Optimal Value**	**Optimal Value**	**# of Robots**	**Configuration**	**Battery reduction idle [%/sec]**	**Battery reduction empty [%/sec]**	**Battery reduction loaded [%/sec]**	**Charging rate [%/sec]**	**Battery level threshold [%]**
% Mishandled Bags	Between 0 and 0.02%	0.00%	26	Six Vertical Areas	0.00328	0.00528	0.0094	0.00391	40.2
			34	Four Horizontal Areas	0.00119	0.0055	0.00258	0.00339	49
			46	Four Horizontal Areas	0.00576	0.00719	0.00685	0.0033	7.7
			51	Six Vertical Areas	0.0097	0.00127	0.00089	0.00921	16.3
			57	Six Vertical Areas	0.00762	0.00075	0.00211	0.00066	70
			59	Six Vertical Areas	0.004	0.00807	0.0081	0.0026	81.6
			78	Four Horizontal Areas	0.0089	0.0088	0.00843	0.00198	27.6
			86	Six Vertical Areas	0.00192	0.00617	0.00551	0.00944	21.9
			89	Six Vertical Areas	0.00379	0.00495	0.00568	0.00897	66.7
			91	Four Horizontal Areas	0.00338	0.00274	0.00523	0.00547	34.8

K.5. Descriptive Statistics Experimental Design III, Scenario: 1624 bags

Table K 5 - Descriptive Statistics Experimental Design III for the 1624 Bags Scenario

Experiment Design III Scenario: 1624 bags	**Average process time of bags [sec]**	**Average % Mishandled Bags [%]**	**Average % empty trips [%]**	**Average % loaded trips [%]**	**Average % charging time [%]**	**Number of conflicts avoided [#]**	**Number of handled bags in runtime [#]**
Number of robots: 1 - 25 Layout configuration: Central Areas							
Minimum	90.27	0.00	57.26	39.08	9.20	0	341
Maximum	13401.29	83.35	60.92	42.74	26.73	1290	1624
Average	1510.98	19.19	58.51	41.49	17.85	990.68	1496.4
Standard Deviation	3310.58	29.95	1.14	1.14	5.79	378.55	321.67
Number of robots: 1 - 25 Layout configuration: Four Horizontal Areas							
Minimum	69.14	0.00	54.71	41.22	8.80	0	359
Maximum	13431.32	82.63	58.78	45.29	25.36	1688	1624
Average	1415.82	17.92	56.17	43.83	17.49	1197.52	1504.48
Standard Deviation	3251.57	29.33	1.25	1.25	5.99	483.48	311.53
Number of robots: 1 - 25 Layout configuration: Six Horizontal Areas							
Minimum	62.37	0.00	51.04	47.35	8.25	0	412
Maximum	12413.33	80.94	52.65	48.96	28.59	1662	1624
Average	1177.74	15.39	51.61	48.39	16.96	1143.36	1522.2
Standard Deviation	2906.64	27.29	0.53	0.53	6.44	479.36	284.49
Number of robots: 1 - 25 Layout configuration: Six Vertical Areas							
Minimum	68.96	0.00	54.77	42.00	8.85	0	366
Maximum	13375.16	82.46	58.00	45.23	25.73	1722	1624
Average	1389.11	17.89	56.33	43.67	17.59	1272.56	1506.12
Standard Deviation	3197.46	29.41	1.17	1.17	6.03	519.01	308.08

K.6. Descriptive Statistics Experimental Design III, Scenario: 4290 bags

Table K 6 - Descriptive Statistics Experimental Design III for the 4290 Bags Scenario

Experiment Design III Scenario: 4290 bags	**Average process time of bags [sec]**	**Average % Mishandled Bags [%]**	**Average % empty trips [%]**	**Average % loaded trips [%]**	**Average % charging time [%]**	**Number of conflicts avoided [#]**	**Number of handled bags in runtime [#]**
Number of robots: 1 - 96 Layout configuration: Central Areas							
Minimum	263.49	8.72	57.36	38.98	6.37	0.00	340.00
Maximum	14278.18	92.69	61.02	42.64	28.12	5589.00	4290.00
Average	1208.90	21.85	57.99	42.01	14.84	4823.15	4024.60
Standard Deviation	2691.65	20.67	0.94	0.94	6.94	1329.42	799.56
Number of robots: 1 - 35 Layout configuration: Four Horizontal Areas							
Minimum	82.10	0.00	55.26	41.09	16.84	0.00	360.00
Maximum	13544.57	92.30	58.91	44.74	28.44	8264.00	4290.00
Average	2469.99	29.25	56.81	43.19	23.07	5574.66	3606.03
Standard Deviation	4013.65	32.98	1.08	1.08	2.65	2799.05	1170.22
Number of robots: 1 - 35 Layout configuration: Six Horizontal Areas							
Minimum	69.16	0.00	51.40	47.09	15.81	0.00	411.00
Maximum	13711.89	91.32	52.91	48.60	28.60	8371.00	4290.00
Average	2185.46	25.92	52.04	47.96	22.85	5474.06	3685.14
Standard Deviation	3847.19	32.24	0.48	0.48	3.22	2774.15	1112.94
Number of robots: 1 - 35 Layout configuration: Six Vertical Areas							
Minimum	81.83	0.00	55.40	41.77	16.88	0.00	363.00
Maximum	13670.82	92.24	58.23	44.60	28.30	8527.00	4290.00
Average	2450.08	28.83	56.91	43.09	23.13	5769.71	3603.69
Standard Deviation	3966.68	32.81	0.97	0.97	2.67	2940.23	1169.39

K.7. Visualization of Descriptive Statistics Experimental Design III, Both Scenarios

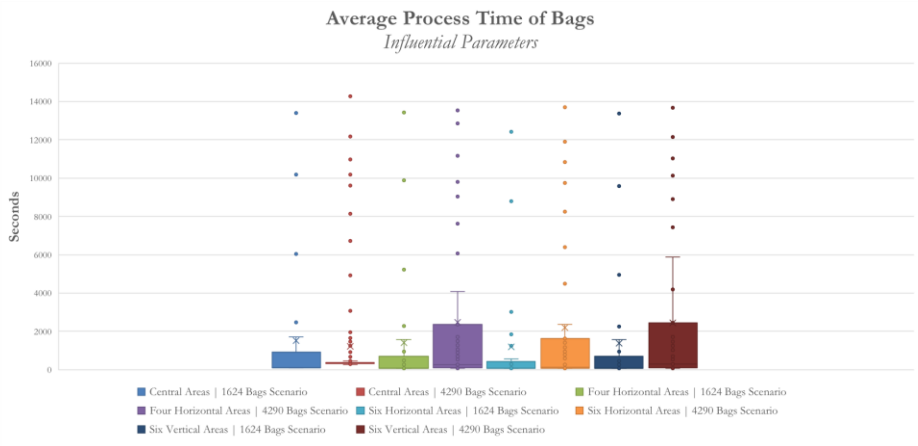

Figure K 5 - Boxplots of Average Process Time of Bags for Experimental Design III

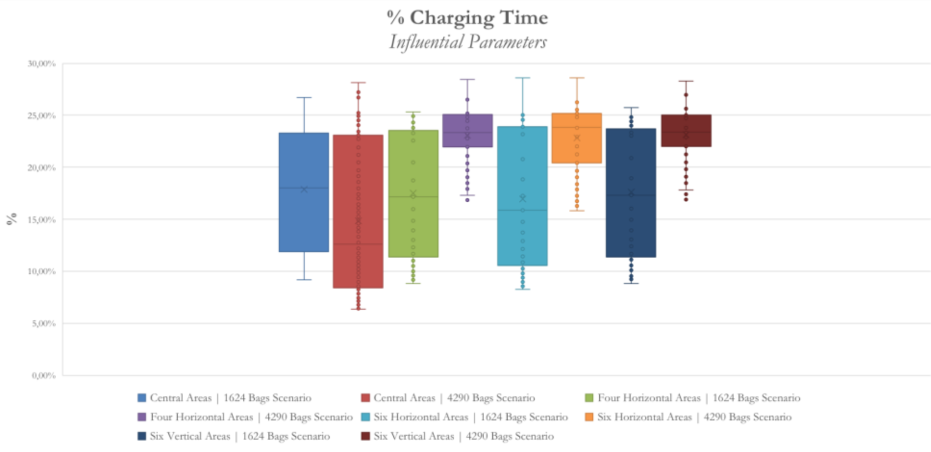

Figure K 6 - Boxplots of % Charging Time for Experimental Design III

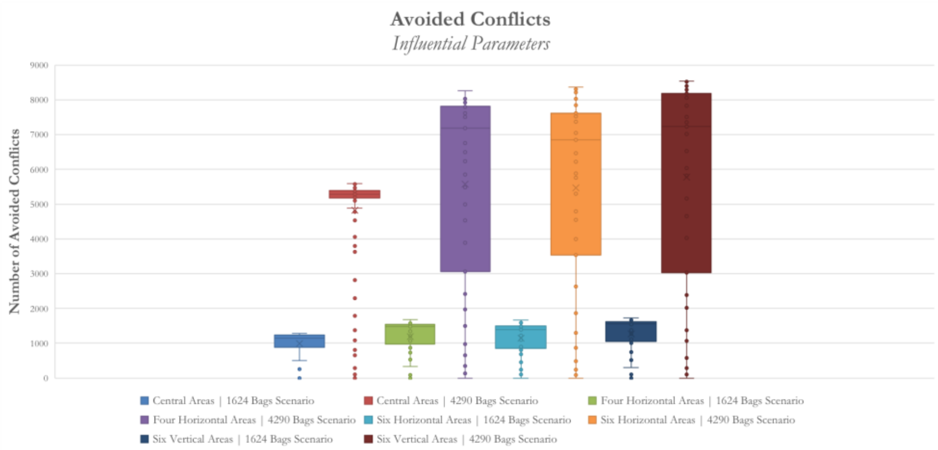

Figure K 7 - Boxplots of Number of Avoided Conflicts for Experimental Design III

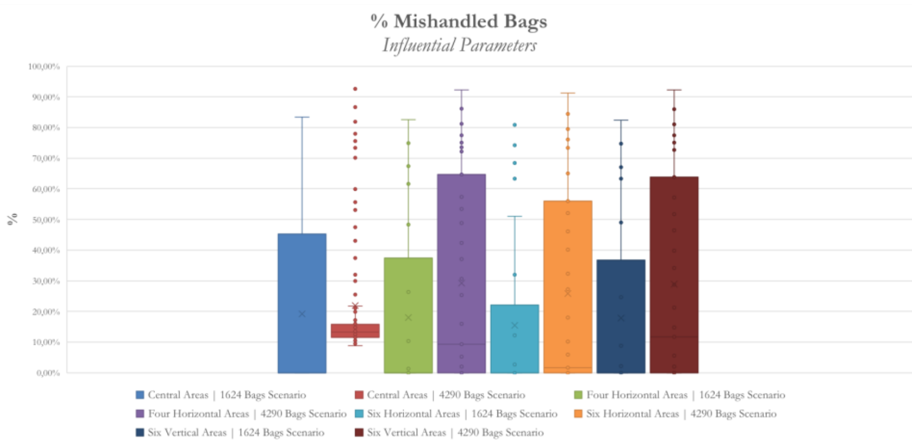

Figure K 8 - Boxplots of % Mishandled Bags for Experimental Design III

L. Using the Model

This appendix includes information on how to use the simulation model as a first time user of the NetLogo software. The focus is on the model interface, which can be used to easily and quickly alter input parameter values and to see the effects of these changes on several output metrics and ultimately the KPI results. The written lines of code corresponding to the model interface are available and commented on in the model file but are not focused on in this appendix.

L.1. Using the Simulation Model

Upon opening the simulation model, the main window with three tabs appears: Interface, Info and Code. In the interface tab the model is visible and it includes tools that can be used to inspect and alter what the model does. Figure L 1 shows the interface of the model developed in this research.

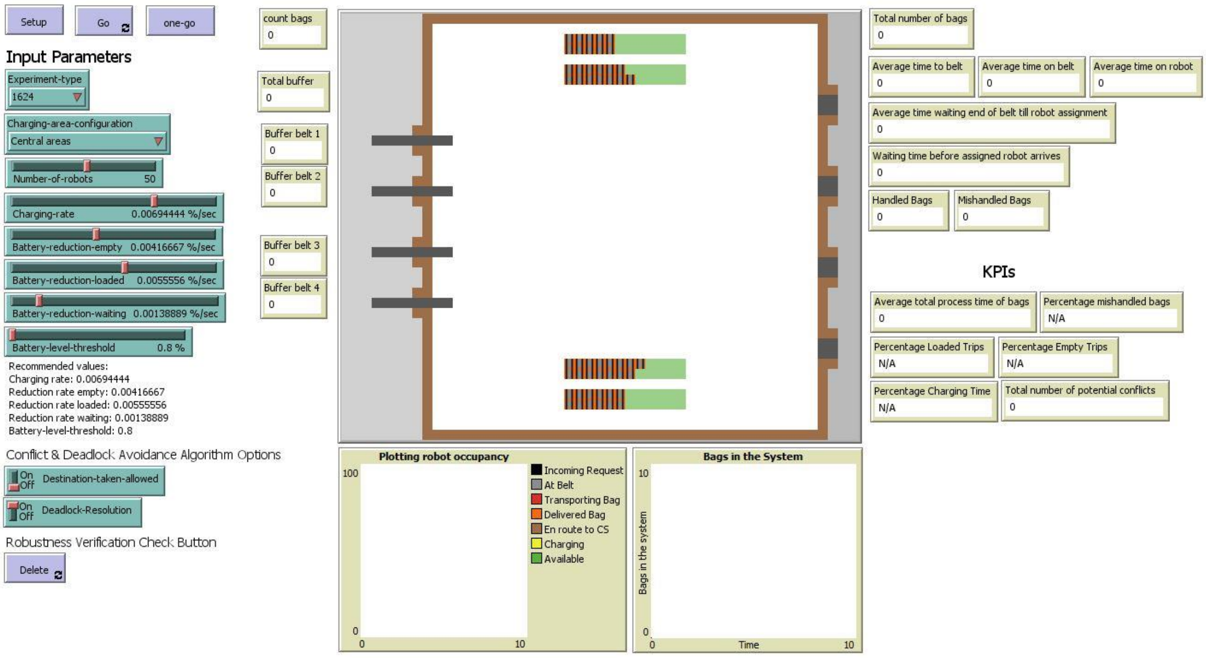

Figure L 1 - Interface of the Simulation Model

As visible, the interface shows several different tools on the left of the model view, plots below the model view and monitors on the right. This section discusses these tools, plots and monitors that can be used by the model user.

L.1.1. Model Input

The NetLogo software allows the model user to use multiple tools to adjust the model input. Amongst these tools are buttons, sliders, switchers and choosers. The inputs for the model are coloured purple and green and are located on the left side of the model view. This section elaborates on the tools used in the simulation model developed in this research.

Setup, Go & one-go

The Setup button on the top left of the model interface sets the model in a state from which it can be run. The button refers to the lines of code in the Code tab of the main window that start with ‘to setup’. In this model, pressing the Setup button resets the world to an initial, empty state, from where it starts setting up the model. In this procedure, the patches are set up, the robots are placed on the green coloured patches and the arrival pattern is set up. Whenever a change is made in one of the input parameters, the user has to press the Setup button before running the model (activating the Go button) to run the model with the changed input parameter values.

The Go button next to the setup button refers to the lines of code in the Code tab of the main window that start with 'to go'. Activating the Go button calls all the procedures that need to start in the model. In this model, these procedures include the inserting of bags into the model according to a predefined arrival pattern, the calling of robots to pick up incoming bags, the movements of bags to and over the incoming conveyor belts, all robot movements and the connections between the robots and the bags. In this procedure, the output metrics are also constantly updated as all procedures defined in the Go procedure are run through at every tick of the model.

The one-go button does the same as the Go button, except that once activated, the Go button loops forever, whereas the one-go button activates all the procedures in the 'to go' code only once, so for one tick only. Using this button instead of the Go button allows the model user to see the changes in the model view and output metric values tick by tick. This can be helpful during for example debugging and is especially helpful when the user turns the randomness in the model off by changing the code line 'random-seed' to ';random-seed', which inactivates the random generator in the model, ensuring that each model run is exactly the same.

Input Parameters

On the left side of the model view, 8 input parameters are visible which will be discussed in this section. The first two input parameters are so-called 'choosers' that let the model user choose between several options from a drop-down list.

The first chooser is *'Experiment-type'* and has two options to choose from: '1624' and '4290'. These options relate to the number of bags that are inserted into the system during the model run. '1624' represents the scenario in which 1624 bags are inserted and '4290' represents the second scenario, in which 4290 bags are inserted.

The second chooser is *'charging-area-configuration'* and has four options to choose from:

- Central areas
- Six Vertical Areas
- Four Horizontal Areas
- Six Horizontal Areas

These options correspond to the floor layout configuration. Each of these configurations has the same number of green patches – 96 – representing the charging and storage positions for robots.

Input parameters three to eight are sliders. Sliders are commonly used as a quick way to change the value of the input parameter without having to recode the procedure every time. The model user can move the slider to a desired value. For the *'number-of-robots'* input parameter, the choice is limited to a range between 0 and 96 robots. This maximum value of 96 is fixed. It prohibits the model user to select more robots than charging and storage positions available in each of the layout configurations. The input parameters *'charging-rate'*, *'battery-reduction-empty'*, *'battery-reduction-loaded'* and *'battery-reduction-waiting'* have a minimum value of 0 and a maximum value of 0.01 percentage points per second and the slider can be changed by increments of 0.001 percentage points per second. These input parameters represent the rates in which the battery power reduces and this can be different for different states of the robot as explained in section 4.4.4.3. The final slider is *'battery-level-threshold'* and can be altered between 0 and 100% with increments of 0.8%.

In the model view, below the input parameters, recommended values are stated in a note in case the model user unintendedly changes the value of the battery reduction and level sliders.

Conflict and Deadlock Avoidance Algorithm Options
Below the input parameters, two switches are present which can be turned on and off. When turned off, *'Destination-taken-allowed'*, prohibits neighbouring moving robots from simultaneously switching places in case of an imminent frontal conflict. In this case, the conflict avoidance measure 'turn, (wait) and continue' is invoked. When this switch is turned on, this conflict avoidance measure is not invoked and the robots involved can take each other's' position during the tick, they are 'taking each other's destination'.

When turned on, the second switch *'Deadlock-Resolution'* makes sure that robots that are involved in an imminent conflict continuously look for a temporary path to a free patch to re-evaluate its shortest path to its original destination. This switch makes sure that robots continuously re-evaluate their shortest path and start following a new shortest path that does not include moving over occupied patches at that time. The combination of the *'Destination-taken-allowed'* switched turned off (and thus invoking collision avoidance measures) and the *'Deadlock-Resolution'* switch turned off (and thus prohibiting robots from re-evaluating their paths) results in deadlock situations.

Robustness Verification Check Button
The *'Delete'* button can be used for verification purposes. One of the situations where it can be used is to check if bags move over the incoming conveyor belts once the bag in front of them suddenly disappears. It checks of the model continuously adapts to changing circumstances created by the model user. When this button is activated, the model user can click on an agent and make it disappear. This agent ceases to exist and therefore will not execute any further code. When the button is activated it can be used an infinite number of times. To deactivate the button, the model user has to click on the button again.

L.1.2. Model Output
In NetLogo, there are several ways of showing the quantitative results after each tick. In the developed model the most used output tool are monitors. Monitors display the value of a reporter from the model, which can be a variable. Monitors automatically update several times per second, providing near real-time values. Another output visualisation tool is a plot. A plot can show data that the model is generating over time. A plot is also updated automatically several times per second when the Go button is activated, meaning the model is running. This section discusses the several monitors and plots used in the developed model.

Count bags and belt buffer monitors
In between the input parameters and the model view, six monitors are located. The first monitor *'count bags'* continuously monitors how many bags are present in the model. When a bag is created in the model, the value of this monitored is increased by one and when a bag exits the model after it's handled or when the model user deliberately deletes the bag with the *'Delete'* button, the value of the monitor decreases with one. The other five monitors on the left side of the model view concern the buffers of the incoming conveyor belts. The most left patch of each incoming conveyor belt is defined as a buffer patch with an unlimited capacity. When a queue arises at the incoming conveyor belts, NetLogo places bags on top of each other, which makes it impossible for the model user to see how many bags are on one patch. This is why in the entire model, having more than one entity per patch is prohibited, except for the buffer patches. To still give the model user insight in how many bags are in each queue, the monitors *'Buffer belt 1', 'Buffer belt 2', 'Buffer belt 3'* and *'Buffer belt 4'* show how many bags are in the buffer zone of the queue. The monitor *'Total buffer´* shows how many bags in total are located in the buffer zones combined.

Bag monitors
On the top right side of the model view, eight monitors show values related to the bags in the model. The first one, *'Total number of bags',* shows the total number of bags that were present at some point during the model run. Unlike the *'count bags'* monitor, the value of this monitor does not decrease when a bag exits the system or when it is manually deleted by the *'Delete'* button. The next three monitors *'Average time to belt', 'Average time on belt',* and *'Average time on robot'*, show the average time bags spend travelling to the belt, travelling on the belt and travelling on a robot respectively. These monitors are updated every time a bag exits the system after it is handled. Every time a bag exits, the bag specific values for the time variables are used to calculate the new average value for the monitors. The same holds for the monitors '*Average time waiting end of belt till robot assignment'* and *'Waiting time before assigned robot arrives'.* The last two monitors, *'Handled Bags'* and '*Mishandled Bags'* are also updated every time a bag exits the system through one of the chutes to the makeup stations. At the end of each model run or experiment, the value for the *'Handled Bags'* monitor has to equal the number selected in the *'Experiment-type'* input parameter. For example, when the 1624 bags scenario is selected in *'Experiment-type'*, at the end of each complete model run or experiment, the value of *Handled Bags'* should be 1624.

KPI monitors
On the bottom right side of the model view, six monitors show the values for the KPIs defined in this research. The first two monitors *'Average total process time of bags'* and '*Percentage mishandled bags'* are updated each time a bag exits the model. The other monitors update at each tick. These monitors can be used as output metrics when running experiments.

Plots
Below the model view, two plots are visible. The first one plots robot occupancy. Robots can have different statuses, being:

- Incoming request
- At Belt
- Transporting Bag
- Delivered Bag
- En route to CS
- Charging
- Available

The plot on the left shows the number of robots per status. It shows that when the number of robots with status 'Available' decreases, the number of robots with other statuses increases. It could show that over time, certain values stabilize. In the research performed, the focus was on end-of-runtime values. However, plots like this can show more valuable information on the performance of the system over the entire runtime series.

The plot on the right shows the number of bags in the system at every point in the runtime. While the model runs, it shows clear peaks that correspond with the peaks in the defined arrival pattern. It can also show the effect of the number of robots present in the system when the entire time series of a model run is considered.

The two plots show how the system performs during the simulated day, instead of only generating end-of-runtime results like some monitors. Insights derived from plots like these can add value to the evaluation of the system as a whole when every time step over the operational period, like a day, is considered.

L.2. Using the Model for Different Situations

There are multiple ways to use the developed model for different situations. However, adjusting the model beyond the limits build into the model interface takes some more effort. In this section, some alterations to the input parameters are discussed, including how to alter the model in such a way that it generates new results.

Changing the Arrival Pattern of Bags

The current version of the model requires a separate .csv file to generate bags according to a certain arrival distribution. To make changes to the arrival pattern or the number of bags, the data in the used .csv files have to be altered. If that is done, the model user can add a new choice to the '*Experiment-type'* chooser by editing the chooser and then editing the 'to setup' code in the Code tab. The same lines of codes can be used for the new .csv file, only the name of the new .csv file and the name of the experiment type have to be adjusted.

Using the model in other industries

The current model considers only four incoming conveyor belts and four entrances to makeup stations. Following the same procedure as described above, these numbers can be altered. The possible changes explained on the arrival pattern of bags focussed on an airport situation. However, this model can also be used for different industries, like for example warehouses. To use the model for other goods than bags, the arrival pattern can be adjusted if necessary, to match the arrival pattern of goods from for example incoming trucks with inventory that drop off their goods at a warehouse. To visualize different goods than bags, the model user can go to Tools -> Turtle Shapes Editor and adjust or add a new shape. In the model code and interface, one simply needs to replace the words 'bag' and 'bags' by the desired word for the goods the model represents.

Changing the floor layout configuration and maximum number of robots

The current version of the model allows the model user to choose between four layout configurations. The maximum number of robots that can be selected by the slider '*Number-of-robots*' is connected to these layouts. All four layout configurations entail 96 green patches, representing the charging and storage positions. As during the night all robots have to be able to charge themselves for the next days, the number of robots in the system have to be equal to or lower than the number of charging and storage positions. When the model user wants to adjust the layout configuration options, there are several parts of the code that needs to be adjusted.

It starts with the globals. Currently, six global types are defined (CS-area-north1, CS-area-south1, CS-areanorth2, and so on). This enables the user to have six separate charging and storage areas. The next procedure that needs to be adjusted is the 'setup-patches' procedure. This procedure is concerned with the location of the individual charging and storage positions.

If the charging and storage positions and areas are altered and the number of these positions is for exampled increased, the user can manually adjust the maximum allowed number of robots. To do so, the '*Number-of-robots´* slider can be edited and the maximum value allowed can be altered.

Changing the battery charging and reduction rates

The current version of the model allows the model user to change the charging rate, the battery reduction when empty rate, the battery reduction when loaded rate and the battery reduction while waiting (idle) rate between the values 0 and 0.01. To adjust this range, or the changing increment of 0.001, the model code does not have to be adjusted. The user can simply edit the sliders one by one and change the range and increment.