The food bank distribution centre in Rotterdam redesigned

A REDESIGN OF THE STORAGE ASSIGNMENT POLICY

MAAIKE KOOL
MSC MANAGEMENT, ECONOMICS, AND CONSUMER STUDIES
OPERATIONS RESEARCH AND LOGISTICS GROUP
JUNE 2018

WAGENINGEN
UNIVERSITY & RESEARCH

Wageningen University - Department of Social Sciences

Operations Research and Logistics

MSc Thesis

The food bank distribution centre in Rotterdam redesigned

A redesign of the storage assignment policy

Abstract
The purpose of this thesis is to improve the currently used randomized storage policy of the food bank in Rotterdam and to give the people who work in the distribution centre of the food bank (administration employees and forklift drivers) recommendations so that they know how the assignment of products to storage locations can be more efficient regarding the number of expired products, the travel time, the and storage time. In the first part the current situation of the storage policy of the food bank is explained. Existing storage assignment policies (i.e. randomized storage, class-based storage, and correlated storage) are described in the second part, and in the third part the first two parts are combined to come up with applicable scenarios for the food bank. With a discrete event simulation model these scenarios are tested for different performance measures (i.e. expired products, travel time, storage time). The results show that class-based storage based on expiration date increases the efficiency of the food bank when looking at the performance measures 'expired products' and 'travel time'.

June, 2018

Student Maaike Kool
Registration number 931130462080
MSc program Master Management, Economics, and Consumer Studies
Specialisation Management in Life Sciences
Commissioner Food Bank Rotterdam
Supervisor(s) Dr Renzo Akkerman (WUR)
Jan Broeders (Food Bank Rotterdam)
Quirin Laumans (Food Bank Rotterdam)
Examiner/2nd supervisor Dr Rene Haijema
Thesis code ORL-80433

WAGENINGEN
UNIVERSITY & RESEARCH

Acknowledgements

With this thesis, there comes an end to six months of hard work. I am grateful for all the help and support that I received during the research of this thesis. I would first like to thank my thesis supervisor Dr Renzo Akkerman of the Operations, Research, and Logistics department at Wageningen University. I appreciated his willingness to invest time in our meetings. Furthermore I am grateful for the constructive feedback I received.

I would also like to thank Irene Gosselink from the Science Shop at Wageningen University who brought me in contact with the Food Bank in Rotterdam and for the financial compensation.

From the Food Bank in Rotterdam I would like to thank Jan Broeders who was available to think about feasible solutions for the problem of the food bank. Also, I would like to thank Quirin Laumans who was always interested in the progress of the research. Furthermore, I would like to thank the volunteers of the distribution centre of the Food Bank in Rotterdam for their openness and sincerity; the all showed me that they have heart for the food bank.

Finally, I would also like to acknowledge Dr Rene Haijema of the Operations, Research, and Logistics department at Wageningen University as the second reader and examiner of this thesis.

Table of Contents

1 Introduction ... 1
1.1 Problem statement ... 2
1.2 Aim of the study ... 2
1.3 Research questions ... 2
1.4 Methods ... 3
2 Current situation of the food bank in Rotterdam ... 4
2.1 The flow of products ... 4
2.2 The flow of information ... 6
2.3 Decisions ... 7
2.4 Dividing the incoming donations ... 7
2.5 Data needed for distribution centre design ... 7
2.6 Storage locations ... 8
2.7 Scope of this thesis ... 9
3 Assignment of storage locations ... 11
3.1 Storage assignment policies ... 11
3.2 Class-based storage ... 11
3.3 Correlated storage ... 12
4 Design of experiments ... 13
4.1 Distribution centre processes: Scenario design for Rotterdam ... 13
4.2 Performance measures ... 15
4.3 Determination runtime and number of runs ... 16
4.4 Current situation ... 18
4.5 Class-based storage based on expiration date ... 22
4.6 Class-based storage based on product categories ... 22
5 Results and discussion ... 23
5.1 Expired products ... 23
5.2 Time spent on storage location ... 24
5.3 Travel time ... 26
5.4 Class-based storage time ... 27
5.5 Sensitivity analysis ... 28
6 Conclusion and recommendations ... 30
6.1 Conclusion and discussion ... 30

6.2 Assumptions....31
6.3 Future research....32
6.4 Comparison with other studies....33
References....35
Appendices....37
A. Model formulas for all scenarios....37
B. Model formulas that change per scenario....41
C. Model parameters....45
D. Capacity and distance of storage locations....47
E. Simulation model food bank without channels....48
F. Simulation model food bank with channels....49

1 Introduction

Improving the management of food waste is becoming more important in many countries worldwide. Food waste occurs at different stages in the food supply chain, but the retail and consumer stages are the most important ones (Parfitt et al., 2010). If food waste has to be reduced at one of these stages, food banks seem to be a good solution to make use of food waste (van der Horst et al., 2014). The definition of a food bank is as follows: a food bank is a non-profit organization that collects food that would otherwise be wasted and transfer it to charitable food programs. Managing the logistics of a food bank, like the collection of food donations, the handling and storage of food, as well as the distribution, is a fundamental activity (González-Torre & Coque, 2016). In Rotterdam, a food bank production and distribution centre is located where food packages are prepared and transported to distribution points in the city and to food banks in the surrounding municipalities. In addition, from the food bank distribution centre, products on pallets are transported to other food banks in the region and the Netherlands where the products are packaged in crates.

Optimizing the operations of a distribution centre, for a food bank or another company, brings some problems. For example, the overall structure of the distribution centre has to be determined. This implies the flow of materials and flow of information. Also, the equipment selection is important; this determines the appropriate automation level for the distribution centre, and identifies equipment types for storage, transportation and order picking. At last, the storage strategy and the order picking strategy are of major importance for a distribution centre. The storage strategy determines what policy is used to assign products to locations in the distribution centre. For the order picking strategy also different policies can be chosen which depends on the operating principles that are selected (Gu et al., 2010).

Some research has already been done to optimize the operations of a food bank. An example of literature related to food bank internal logistics is from the USA, where a data analysis was done to predict when food donations come in, because these donations are often hard to predict and this makes it for a food bank hard to plan and distribute the donations properly (Jiang et al., 2011). In France, a study was done to optimize the nutritional quality of the food aid that is delivered by food banks. With a linear programming analysis it was showed that changes in the types and amounts of food will improve the nutritional quality of the food that is delivered by the food banks (Rambeloson et al., 2008). Another example is from Spain where food banks are studied from the point of view of business management, their position in the supply chain in which they are integrated, and the relationships upstream and downstream (González-Torre & Coque, 2016). In the USA, a transportation schedule was developed that enables the food bank to collect donations from local sources and to deliver the food to charity agencies. The minimum number of food delivery sites that are necessary to satisfy the agencies was identified in the model (Davis et al., 2014).

These studies are all examples of optimization problems related to food banks. However, no research has been done to the optimization of the internal logistics of a distribution centre of a food bank so that it can be designed in the most efficient way. And more specific, no research has been done to the assignment of storage locations in a food bank. Most optimization literature is concerned with companies that emphasise economic objectives while a food bank is a company that has mostly social and humanitarian objectives. This makes that the optimization of a distribution centre for a food bank is different than for another company, because for food banks the social aspect is more important; they provide food to socially isolated and poor people. For example, for a food bank it is important that the number of expired products is minimized, because otherwise it has to be thrown away which increases food waste and leaves less for the food crates. For another company, it could also be important that the number of expired products is minimized, but the reason to do so is that otherwise it reduces the profit. Also, general speaking, a food bank has more stakeholders and they deal with a greater complexity since they never exactly know when a food donation is coming in (Larson & McLachlin, 2011).

1.1 Problem statement

For the food bank distribution centre in Rotterdam, the donations they receive are coming in on pallets and these pallets are stored on the shelf in the distribution centre somewhere where a place is left; the forklift driver chooses the storage place. Because of this random storage policy products might expire or get lost somewhere in the distribution centre. In other words, the storage assignment policy of the food bank might not be optimal and could be more efficient by using a different storage assignment policy.

1.2 Aim of the study

The aim of this study is to investigate what the current situation regarding the storage assignment policy of the food bank in Rotterdam is. Also, the different potential storage assignment policies need to be determined. These policies are to be assessed for use in the food bank in Rotterdam so that an improved policy can be made. The final aim of this study is to provide recommendations for the food bank and to come up with future research subjects.

1.3 Research questions

The aim of this study leads to the following central research question:
What is the optimal design of the internal logistics regarding the assignment of storage locations of the food bank distribution centre in Rotterdam?

This central research question is divided in four sub questions:
Internal logistics

1. What are the internal logistics requirements for the food bank distribution centre in Rotterdam, in terms of the flow of products, the flow of information and the logistics decisions that are made, regarding the assignment of storage locations?

Redesign of the distribution centre

2. In what way can the assignment of storage locations in distribution centres be organized; what are important elements?
3. What are possible redesign scenarios that fit the distribution centre of the food bank in Rotterdam, regarding the assignment of storage locations?
4. What scenario of the assignment of storage locations is best applicable for the food bank in Rotterdam and what recommendations belong to that?

1.4 Methods

To methods that are used to answer the questions as described in the previous section are described in this paragraph.

For the first question I went several times to the food bank in Rotterdam to speak with the people who work there. Small interviews were held with women who work behind the administration desk, a forklift driver, a man who designed the computer system and someone who works at the recruitment department. The distances between the shelves in the distribution centre were measured approximately and the surface of the different areas like the fridge, the freezer, and the storage area as well. The findings are described in chapter 2.

In chapter 3 different storage assignment policies are described to answer sub question 2. To obtain information about different storage assignment policies a literature study was performed in which the following search terms were used to get useful articles are: storage assignment, policy, strategy, distribution centre, warehouse, design, and combinations of these terms.

For the third question the policies found in chapter 3 were applied to the current situation of the food bank; 6 different scenarios were made. To test the efficiency of these scenarios a discrete event simulation model was made and 3 performance indicators were measured: the number of expired products, the storage time of products, and the travel time in the distribution centre. The different scenarios and an explanation of the model can be found in chapter 4.

The last question is answered in chapter 5 where the results of the simulation model are described and discussed. To give a proper overview of the found results boxplots were made and one-way ANOVA tests were performed to check if the performance measures are really different between the different scenarios.

2 Current situation of the food bank in Rotterdam

The first question that needs to be answered is what is the current situation of the distribution centre (DC) of the food bank in Rotterdam. This includes the flow of products, the flow of information and the decisions that are made. In Figure 1 a schematic overview of the food bank distribution centre in Rotterdam is given. In this figure a floor map of the food bank is given with indications of the different areas.

The surface area of the shelves is about 825 m^2, the freezer exists of 60 m^2 and the fridge in total is 182 m^2. Most of the shelves exist of three levels of approximately 8 meters. A factor that needs to be taken into account is that the shelves in the distribution centre cannot handle too much weight (they can carry a maximum amount of 2000 kilograms), so the heavy pallets need to be stored on the ground. The distribution centre can handle around 500 pallets as a maximum.

2.1 The flow of products

The flow of products is the route that the products travel through the DC. It all starts with a donation of food products from a company. Most of the time the people at the administration desk know at what time a donation is coming in, because a company has contacted them the same day or a few days before the donation is coming in. Sometimes a company brings the donation itself and sometimes the food bank has to pick up the donation at the company. When a donation is coming in, it enters the DC through the front door. It is put on the weighing scale by a forklift truck and information about the product is registered there as well. All this information is printed on a receipt and this receipt is put on the pallet. After this, a forklift truck

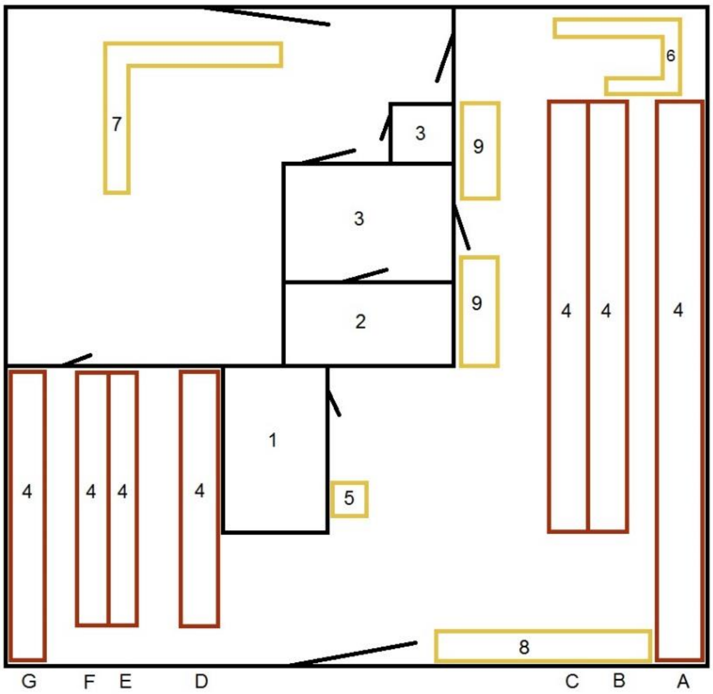


Figure 1 Schematic overview of the foodbank DC in Rotterdam. 1. administration; 2. freezer; 3. fridge; 4. shelf; 5. weighing scale; 6. dishwasher for crates; 7. assembly line; 8. forklift trucks; 9. extra storage. The letters underneath are the different storage locations.

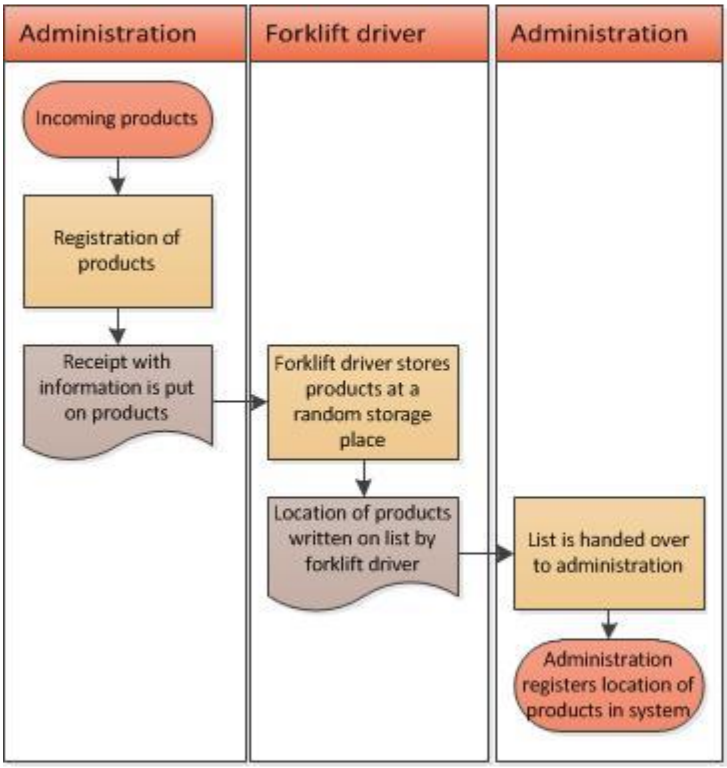


Figure 2 Flow of products from the donation till the administration of the location.

is putting the pallet in one of the shelves. The storage location is chosen by the forklift driver. If there is space left somewhere in the shelves, the pallet is put there. The only criterion that is taken into account when choosing a spot for the pallet is the weight of the product and sometimes the product category as well; products from the same product category are stored together. When products are coming in they are assigned to a specific product category which are: potatoes, vegetables, and fruit; bread; eggs; grain and rice; coffee, tea, juice, and lemonade; groceries; fish and meat; dairy products; and other. The shelves cannot handle too much weight, so the heavy products are stored on the ground underneath the shelves. After the pallets are stored, the fork lift driver gives the location of where he stored the pallet to the administration and they save the location in the system so that it is easy to find the location of the pallet again when it is needed. In Figure 2 this process is shown in a flowchart.

The crates with food products are made by volunteers every week from Monday till Thursday at the assembly line. For these crates, food products are necessary and these need to be selected from the shelves. A list with all the products that are needed is printed at the administration desk and given to the fork lift drivers so that they know which products they need to take out of the shelves. They collect the pallets and bring them to the beginning of the assembly line where the food crates are made. The finished crates are stored in the large fridge until they are transported to the local food banks. When the crates return from the people who got one, they are washed in a dishwasher. This process is shown in Figure 3. In this figure, two alternative paths are shown. For the first path the collector of the food products is choosing the products and informs the administration which products are chosen. With the second path, the collector is choosing products from the shelves, but the administration is not properly informed about which products are chosen.

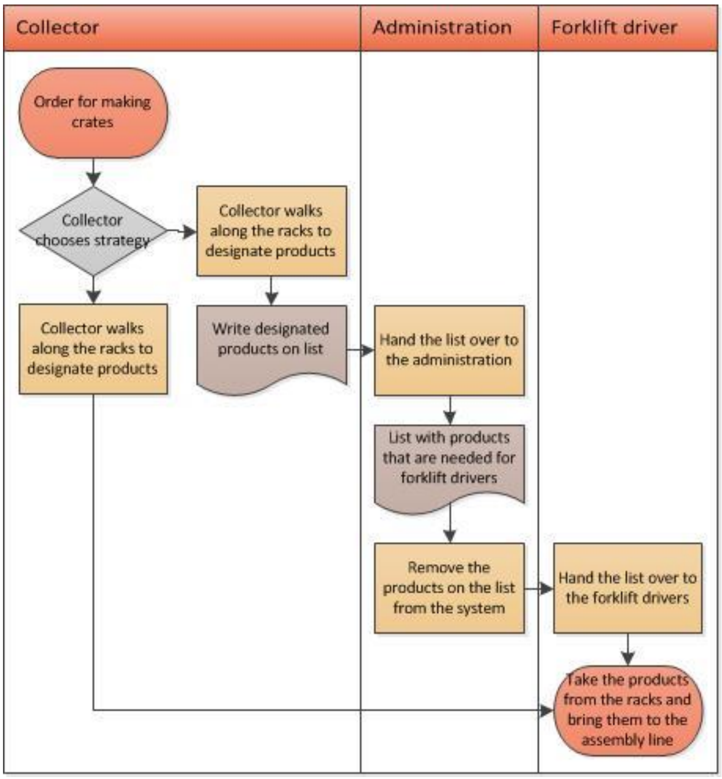


Figure 3 Flow of products from the order till the delivery at the assembly line.

2.2 The flow of information

The flow of information is about all the information that is collected about the incoming donations and the information that the volunteers are giving to each other.
The information that is registered about the pallet with products is: the number of pallets, the product that is on the pallet, the number of products on the pallet, the weight of the pallet, the package of the product, the expiration date of the product, the date when the product is stored in the DC, the arrival temperature of the product, the supplier of the product, and a number is given to the pallet. All this information is registered on a receipt and attached to the pallet. On this receipt a QR-code is printed as well. This code can be scanned to assign a storage location to the pallet automatically, but this code is not used, because it costs too much to buy new tools to scan the codes.

When the pallet is stored, the storage location is given to the administration by the fork lift driver, this happens manually. Sometimes mistakes are made in this phase, because people do not always pass along the right storage location of a stored product.

A few people at the food bank are responsible for the selection of the food products. They make a list of products that are necessary for the crates, based on the expiration dates of the products and the diversity and healthiness of the food crate. When they have selected which products will be used the administration is making a list of the products that will be taken out of the shelves and this list is printed. The products are then also removed from the computer system. However, sometimes the collector is walking along the shelves and is pointing at the products that he wants to have for the crates without properly monitoring which products are taken. In this situation the administration does not know which products are used and next time when crates are made there is a miscommunication between the person who choose the products and the administration, because the administration thinks that products are still in stock while in reality these products are already used for previous food crates. As described before, this process is shown in Figure 3.

2.3 Decisions

The people who decide on which products are accepted from donors are working on the department of recruitment. They ask companies for donations and decide whether or not a donation is accepted if a company is offering one.

The decision where to store the pallets is made by the forklift driver. He determines where a product is stored and on which characteristics this is based. This can be random and sometimes based on product categories; products from the same product category are stored together..

To decide which products are going in the crates two persons are responsible. They decide on which products are taken and which are not so that a varied crate is formed and also the expiration dates are taken into account. The expiration date is a large influencer in the time that a product is stored in the shelves. Sometimes donations are coming in that have an expiration date of the next day, so they need to be used immediately, while other donations, for example canned vegetables, have a date that expires in more than a year so they stay much longer in the DC. However, sometimes expiration dates are not managed well and products still need to be thrown away.

2.4 Dividing the incoming donations

At the food bank in Rotterdam donations are coming in, and these donations are used to fill the food crates, but products that are not used for the crates are transported to other food banks or distribution centres in the region. Some companies that donate products assign these products for nationwide use instead of for Rotterdam only. When this happens, part of the donation can be used for the food crates in Rotterdam, but the other part needs to be transported to other food banks or distribution centres. When a donation is not assigned by a company it can be used completely in Rotterdam, but it can also be transported as well. However, to investigate whether the assignment of products to different regions has an effect on the storage assignment policy of the food bank is out of scope for this thesis.

2.5 Data needed for distribution centre design

To redesign a distribution centre data about the stock keeping units (SKU) is necessary, because the inventory determines the size and capacity of the stock, the flow of the facility and the assembly areas within the distribution centre. Important aspects of SKU data are: the number of active SKU's handled currently, the number of SKU's to be handled in the future, the design specifications of the SKU's that are handled and the storage requirements.

Another important aspect of designing a distribution centre is the item master data. This is data that ensures that the facility can adequately store and handle the items that are processed. Item master data that should be known is: a description of each physical item, number of each item generally kept in stock, the type or category for each item, dimensions

(length, width, height, weight) for each item, unit of measurement, products per pallet level and the number of pallet levels on a pallet. Currently, the food bank is recording this data well.
The design of a new distribution centre relies on accurate order history of the current distribution centre. Important order history data is: order number, date and time that the order was picked, items in the order, quantity of items in each order, when an order was placed, unit of measurements in an order. It is best to include data for the last year's orders (Ciervo, 2017). Also this information is registered correctly by the food bank.

2.6 Storage locations

There are currently seven storage locations in the distribution centre in Rotterdam. I divided these seven locations in half, so that fourteen locations occur. I did this, because the shelves are quite long and the difference between the front and the back of the shelves is too big to take it as a whole. In Figure 4 the different storage locations are shown.

All the shelves exist of smaller sub shelves which can carry 12 to 16 pallets. The only exception is shelf A which can carry 9 to 12 pallets, this is because shelf A has only two levels instead of three. The number of pallets that are actually stored on a shelf depends on the number of products that is on the pallet which determines the total size of the pallet. If there are a lot of products on a pallet its size is bigger which has results in less pallets per shelf.

In Table 1 more information about the different storage locations is given. It includes the number of shelves per storage location, the average number of pallets that can be stored per storage location, the distance from the assembly line to the storage location, and the rank of the storage location based on the distance from the assembly line. The rank of the storage locations is necessary to identify what the most preferable storage location in the distribution

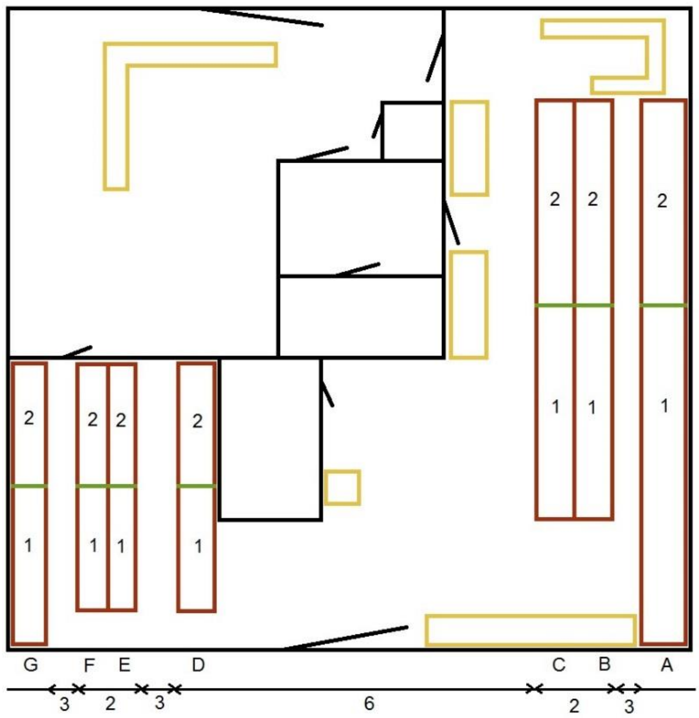


Figure 4 Distribution centre of the foodbank in Rotterdam with the different storage locations A1 till G2. Also the distances between the shelves is given underneath the picture. The green lines determine where the shelf is cut in half.

Table 1 Information about the different storage locations. In the last column the most preferable and the least preferable locations are stated.

Storage location	Number of sub shelves	Number of pallets that can be stored in the shelves in total	Distance from assembly line till shelve (m)	Rank based on distance from assembly line
A1	4	40	25	4
A2	5	50	41	11
B1	2	28	22	3
B2	3	42	30	7
C1	2	28	20	2
C2	3	42	12	1
D1	3	42	26	5
D2	4	56	38	8
E1	3	42	29	6
E2	3	42	41	11
F1	3	42	31	9
F2	3	42	43	13
G1	3	42	34	10
G2	4	56	46	14

centre is. I measured the distances in the distribution centre approximately by counting the number of footsteps it takes. I assumed that when I took a big step it was one meter. The distance between the shelves is shown in Figure 4 as well. The green line in the shelves is determining where the shelf is cut in half to create more storage locations.

2.7 Scope of this thesis

With all the current processes and decisions of the food bank distribution centre described it is necessary to define the scope of this thesis, because it is not possible to include everything in this thesis. In Figure 5 an overview of all processes and decisions is shown. The yellow rectangles represent the processes of the foodbank from when products are arriving at the distribution centre till they arrive at the assembly line. Of course, there are processes before and after the arrival at the distribution centre and the assembly line, but it is too much to describe every process and decision in the whole chain. The grey diamond shaped figures represent the decisions that are made as described in paragraph 2.3. Also two feedback paths are shown. The first one is from 'storage location' to 'administration' and represents the communication of the storage location from the stored products by the forklift driver. The second feedback path is from 'retrieve products' to 'administration' and represents the communication of which products are retrieved from the shelves when the food crates are made by the people who are responsible for compiling the food crates.

In the big black rectangle the processes and decisions that are within the scope of this thesis are shown. These processes start when the fork lift driver is deciding where to store a specific product and end when the forklift driver has taken the products to the assembly line.

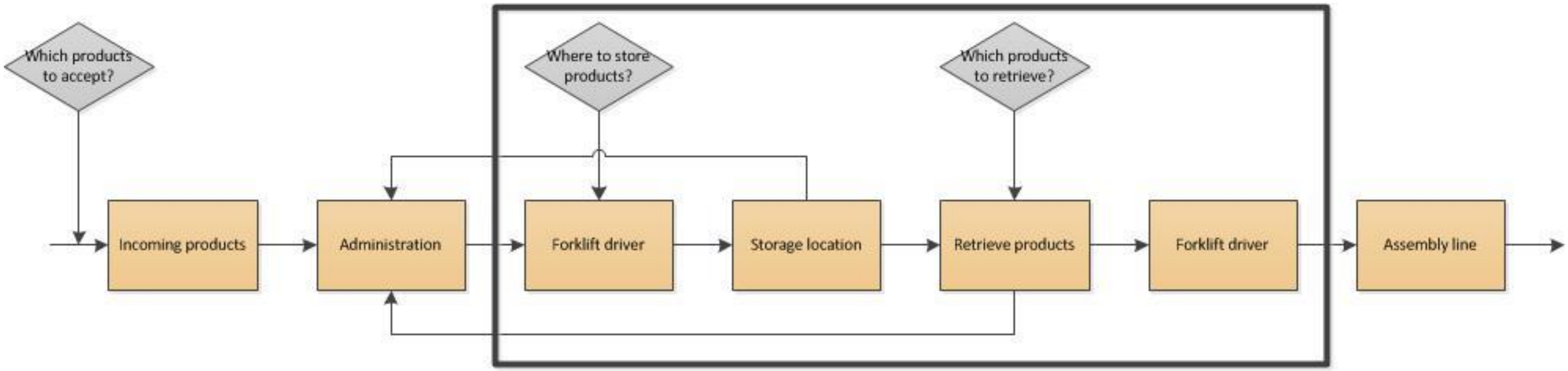


Figure 5 An overview of the different processes and decisions at the distribution centre of the food bank in Rotterdam. The processes are shown in the yellow rectangles, while the decisions are shown in the grey diamond shaped figures. There are two feedback paths; one from 'storage location' to 'administration' and one from 'retrieve products' to 'administration'. The first one is the communication of the storage location to the administration and the second one is the communication of which products are retrieved to the administration. In the black box the processes and decisions that are in the scope of this thesis are shown.

3 Assignment of storage locations

In this chapter different storage location assignment policies as found in literature are discussed. The main processes of a distribution centre are receiving, storage, order picking, and shipping. Receiving is about the arriving of a product, storage about the products placed in storage locations, the order picking refers to the retrieval of the products, and the shipping is about the orders that are checked and loaded in trucks (Rouwenhorst et al., 2000). In this thesis, I will mainly look at the storage process of the distribution centre, because the problem of the storage location assignment lies within that area.

In more detail, the storage location assignment procedure is as follows: the first point is how products are distributed among warehousing systems. This is because most large distribution centres contain more than one type of warehousing system. Each system is especially equipped for a specific group based on characteristics like size or weight. For the food bank in Rotterdam the weight of the products is important, because the heavy pallets need to be stored on the ground. The second point of the procedure is the clustering of correlated products. This means that products that are often ordered together are also stored at the same place so that the travel times for order picking may reduce. The third point of the procedure is that the workload within a warehousing system is balanced. The throughput capacity may increase by distributing products among different zones such that the workload is balanced among zones. The fourth and last point of this procedure is the assignment of products to storage locations (van den Berg, 1999). For assigning products to storage location several storage policies exist. A storage policy determines how storage locations are allocated to products; it involves the transfer of the incoming products to the storage location (de Koster et al., 2007).

3.1 Storage assignment policies

There are several frequently used types of storage location assignment. For the first one, *dedicated storage*, all products are stored at fixed locations. Every product, or pallet with products, can be found in the same location for a longer period of time. In *randomized storage* a product is assigned to a current available, empty location. This means that the same products will be found in different locations each time (Wäscher, 2004). With randomized storage people can choose the storage location themselves, and it is likely that a system called *closest open location storage* will occur. This is that the first empty location that is encountered will be used to store the product.

3.2 Class-based storage

A combination of some of the methods mentioned so far is called *class-based storage*. The idea of this policy is that products are grouped into classes in such a way that the fastest moving class contains about 15% of the products stored, but contributes to about 85% of the turnover. An advantage of class-based storage is that activities are evenly distributed over the distribution centre and this has the consequence that obstruction in the distribution centre is reduced and because of this the throughput capacity is increased (van den Berg & Zijm, 1999). Each class is assigned to an area of the distribution centre, but storage within an area is random (de Koster et al., 2007). To create such classes, two questions need to be answered.

The first one is how many classes are used and the second one is how the borders between the classes are determined. An approach to classify products is the ABC analysis. This technique classifies product groups based on demand value or demand volume based on the Pareto principle. For example, the products in group A are generating 70% of the company's revenue but only taking up 10% of the inventory. The products in group B are generating 20% of the company's revenue and taking about 20% of the inventory and the products in group C are the products that generate only 10% of the company's revenue, but taking up about 70% of the inventory (Ng, 2007). Another approach to divide products in classes is the fast, normal, and slow moving technique. This technique distinguishes product classes based on the demand rate (van Kampen et al., 2012). Also the cube-per-order index (COI) is used to assign products to classes. The COI is the ratio of storage space that is required for a product to its popularity. It ranks products based on their COI values, and after the ranking products are allocated in ascending or descending order to the most accessible locations (Malmborg & Bhaskaran, 1990).

3.3 Correlated storage

Relations between products that are stored may exist. For example, a customer may tend to order a certain product together with another product. Or, applied to a food bank, when pasta is put in a food crate, it is likely that pasta sauce is also needed. This is called *correlated storage* or *family-grouping*; similar products are located in the same region of the storage area (de Koster et al., 2007). Correlated storage assignment takes the correlation among products into consideration to find more economical solutions to increase the efficiency of the retrieval of products (Zhang, 2016), because with correlated storage the travel time in a distribution centre may be reduced (van den Berg, 1999). To find out which products are correlated, historical data about product registration can be used. With cluster analysis groups of items that customers frequently order, or groups of items that are used for food crates, can be identified. Products with high values of correlation are then stored near to each other (Bindi et al., 2009). The correlated storage strategy consists of two sub-problems: the identification of clusters of products and the assignment of the clusters to locations in the distribution centre (de Ruijter, 2007).

4 Design of experiments

To model the activities of the food bank a simulation model is used. In simulation, a model of a real-life system is created that describes processes involving individual units such as persons or products. In a simulation model it is tried to reproduce the actual operations of the real situation (Bangsow, 2012). It is useful to use a simulation model when someone wants to get more insight in a current situation. In this chapter, scenarios based on the previous discussed storage assignment policies are explained first. Next, the performance measures and the determination of the number of runs and the runtime is discussed. In the last part of this chapter the simulation model is explained in more detail.

For modelling the distribution centre in Rotterdam I will use discrete event simulation (DES). For this kind of simulation events happen at discrete instances in time. The events take zero time to happen and it is assumed that nothing happens between two consecutive events (Tako & Robinson, 2010).

Before I started to build the model I received a database from the food bank which contained all the incoming donations from 2017 and information about these donations. The information that is registered about the donations is: pallet number, the date a donation comes in, the date a donation goes out, donation received from which company, main product group, food group, product, expiration date, number of products per package, package material, total number of products, weight, temperature, location, destination, reserved for. Not all parameters are always filled in when a donation is registered. The package material, the temperature and the location in the distribution centre is often left blank in the dataset.

The number of data points in the dataset is 7411 donations. However, from these points some of them were not usable, because important information was missing such as the expiration date. Also, this set of points contained products that needed to be stored in the fridge or the freezer. Since it is not possible to think of a storage policy for the fridge or freezer, because those rooms are too small, I decided to delete these from the dataset. In total, the number of points I deleted was 947, so that left me with 6464 usable data points.

The simulation model was made with simulation software called Enterprise Dynamics 9® version 9.0.0.1342 on a computer with Windows 7 Enterprise. In the simulation software a tool called 'Autofit' was present. With this tool the distribution of a dataset is calculated and a score is given to this distribution. The score indicates how good the distribution fits the data. A condition is that the score has to be at least lower than 100, but the lower the score, the better the distribution. This tool is used to fit the distributions of the model; if it used it is mentioned in the relevant paragraph.

In Appendix E and F the complete simulation model is shown and in Appendix A and B the formulas that are explained in this chapter are shown in more detail.

4.1 Distribution centre processes: Scenario design for Rotterdam

The scenarios that are selected for the distribution centre of the food bank in Rotterdam are based on the location assignment policies as described in chapter 3. In the end there are six different scenarios including the current situation of the food bank.

4.1.1 Current situation

The first scenario is the current situation of the food bank; it is the scenario as described in chapter 2. The focus in the model is on the assignment of products to different locations and on how the products are retrieved. However, the processes before the storage and after the retrieval are also shown for the completeness of the model. Modelling the current situation is necessary to have a benchmark to compare the other scenarios with. In the current situation *randomized storage* is used, so a storage location is randomly chosen and also the retrieval of the products is random; the expiration date or other characteristics are not taken into account. As an extra scenario the model can be adjusted in a way that the expiration date is taken into account; the products with the shortest expiration date are taken first from the shelves.

4.1.2 Class-based storage based on expiration date

For the second scenario *class-based storage* is used. The structure of the model is the same as for the current situation, but for this one the products are divided in classes. As mentioned before, classification of products is based on demand value or demand volume and the goal is to reduce the mean travel time for storage and order picking. However, for the food bank in Rotterdam, other characteristics are more important than the demand value or demand volume. For example, the expiration date/perishability, products that are used a lot, weight of products, shape of the products, size of the product, or the healthiness of the product, et cetera (van den Berg, 1999).

The first step in class-based storage is that the number of classes needs to be defined. The classes are defined based on the expiration date of the products. Based on ABC classification the default of threes classes is chosen.

The second step is to determine the borders between the different classes. For example, in case of three classes the products with a short expiration date will be stored in class A, because they are needed soon, the products with an intermediate expiration date will be stored in class B, and the rest of the products, with a long expiration date will be stored in class C. Class A is preferably close to the entrance of the distribution centre or close to the assembly line. Class C is somewhere far from the entrance or the assembly line, and class B is somewhere between class A and class C. To define the borders of the classes more specifically; if a product has an expiration date of 30 days or less it goes in class A, if the expiration date is between 30 and 70 days it goes in class B, and if the expiration date is over 70 days the product goes in class C.

As described, the storage of the products is based on the expiration date, but for the retrieval of products there are two options. The first is that the products with the shortest expiration date are taken first and the second option is that the retrieval of products is based on which product came in first, just as in the current situation scenario.

4.1.3 Class-based storage based on product categories

For the third scenario, class-based storage based on product categories is used. The difference with the previous scenario is that for this one the number of classes is already defined, because these are the different product categories. The food bank is already assigning

Table 2 Overview of the designed scenarios. The storage policy and the outgoing policy are given. FIFO: 'first in first out', FEFO: 'first expired first out'.

Scenario	**1**	**2**	**3**	**4**	**5**	**6**
Storage policy	Random	Random	Class-based (expiration date)	Class-based (expiration date)	Class-based (product category)	Class-based (product category)
Outgoing policy	FIFO	FEFO	FIFO	FEFO	FIFO	FEFO

all products to different product categories so this is convenient to use as classes. If different products are stored together this can have an advantage in finding the products back, because they are stored in a more logical way. The food bank is already dividing incoming donations into categories which are: potatoes, vegetables, and fruit; bread; eggs; grain and rice; coffee, tea, juice, and lemonade; groceries; fish and meat; dairy products; and other. So, the products are categorized, but this categorization is not used to assign the products to a specific storage location yet. Some of these products are stored in the fridge or freezer. I removed those groups from the list, because the fridge and freezer are too small to have a storage location assignment policy. For convenience I deleted the groups fish and flesh and dairy products from the database. This results in seven groups that are left. Based on the number of products in the groups they are sent to different locations in the distribution centre. Within these groups the products can be ordered based on the expiration date, so that the products with the shortest expiration date are taken first, or they can be ordered based on which product came in first.

An overview of the different scenarios and the different outgoing policies can be found in Table 2. As explained in paragraph 2.2, when products are retrieved from the shelves they are selected based on their expiration date. However this is not always the case and that is why there are two different outgoing policies used for the three different storage assignment policies. First in first out (FIFO) means that the product that comes in first is going out first as well. First expired first out (FEFO) means that the product that is first expired is going out first.

4.2 Performance measures

For the designed scenarios different performance measure can be measured. In class-based storage the demand value is often the leading performance measure. However, for the food bank demand value is not important because all the incoming donations are a gift and the products are not sold to people, but handed out for free. Other performance measures that are often relevant for simulation models are: throughput time, the number of requests that are handled per time period, the total time required to handle a certain number of requests (Roodbergen & Vis, 2009), average production time, and percentage of working time (Ghanmi, 2006). Also the costs of new investments that are necessary for a new design are important, although this is not really a performance measure.

There are three performance measures that are used for testing the scenarios. The first one is how many products are expired when packed for the crates. With products the pallet with a number of products on it is meant. Some products can be handed out when they are expired, but other products cannot. The percentage of expired products is preferably as low as possible, because then food waste is reduced.

The second performance measure is the distance travelled in the distribution centre. With a decrease in the travelled distance the efficiency of the distribution centre can be increased, because it takes less time to store and retrieve the products.

The third performance measure for the food bank is the total time that a product was stored in the distribution centre. If the time that a product stays in the distribution centre is reduced, there is less chance a product is expired.

4.3 Determination runtime and number of runs

For the simulation model the runtime and the number of runs needs to be determined. To test if the parameters I defined are reliable I did some test runs with different runtimes and number of runs. The default settings are a runtime of 365 days and 500 runs. For a second run the runtime was divided in half, so 182.5 days and the number of runs was still 500. A third run exists of 365 days and 250 runs and for the fourth test run the settings were 250 days and 500 runs. All the runs were done without a warm-up period. The performance measure that was tested was the storage time in days of the products and scenario 1 was used. It is too much to test every scenario and scenario 1 is the current situation of the food bank and used as benchmark. The results of the four runs is shown in Figure 6 in a boxplot. It can immediately be seen that there is a lot of difference between the different runs. In fact, a one-way ANOVA test shows that most runs are significantly different from each other. The significance level that is used is $\alpha = 0.05$. With a P-value of 0.000 it is shown that all scenarios have a different mean, except for scenario 1 and 3 that have a P-value of 0.962.

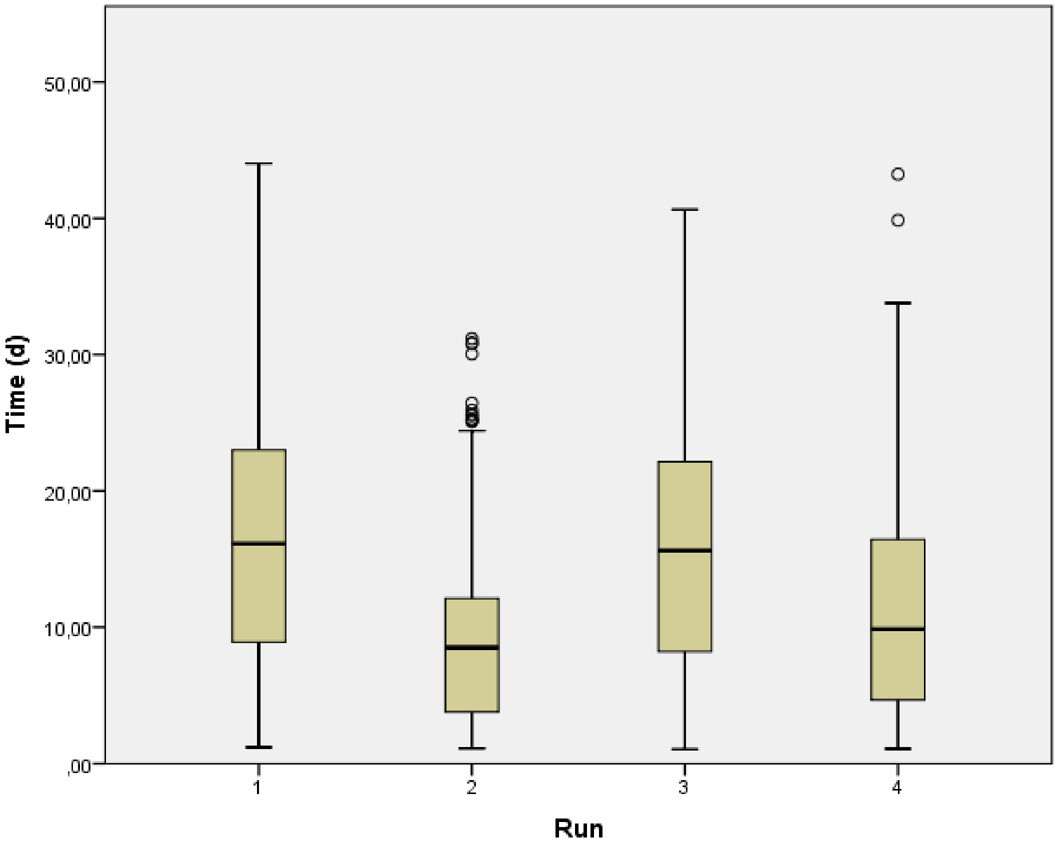


Figure 6 A boxplot of the storage time in days for scenario 1 for different runtimes and number of runs. 1: 365 days, 500 runs; 2: 182.5 days, 500 runs; 3: 365 days, 250 runs; 4: 250 days, 500 runs.

As can be seen in Figure 6 a steady state of the model is not reached. This means that the inflow and outflow of the model are not balanced. In fact, the inflow of the model is lower compared to the outflow. This has as a consequence that the results of the simulation model might be influenced by the fact that there are not enough products in stock. For example, if there are not enough products in stock and an order is coming in, every new incoming product is directly used for the order and this results in less expired products. To reach the steady state of the model the inter arrival time is adjusted. In the first 24 hours of running the model 200 products come in, so that the distribution centre is already filled with products and not empty if the first order comes in. After this 24 hours, a uniform distribution is used which has on average the same input of products as that the order generator makes on output. The warm-up period of the model was adjusted to 504 hours which is 21 days. This is to make sure that the 200 products that are incoming in the first 24 hours left the model without being counted. Their expiration date could have a negative and unreliable influence on the end result. To test if these new setting are better compared to the old one the same runs were done and the results of these runs are shown in Figure 7. At first sight these runs look far more similar compared to the ones in Figure 6. With a one-way ANOVA was tested if this is really the case. The P-value of this test is 0.069 which says that there is no difference between the different groups. As a result it can be said that the new system parameters were adjusted to assure that the simulation model runs in a steady state. The final settings for the runs of the model are set to 250 days and 500 runs.

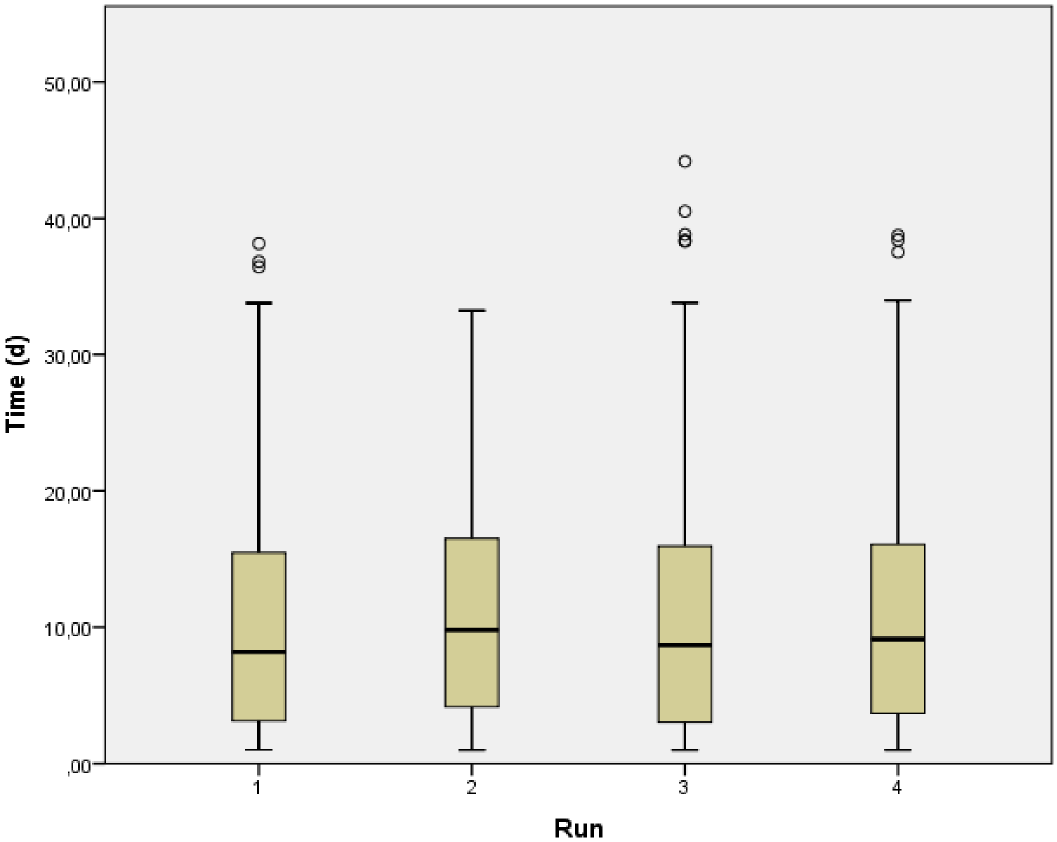


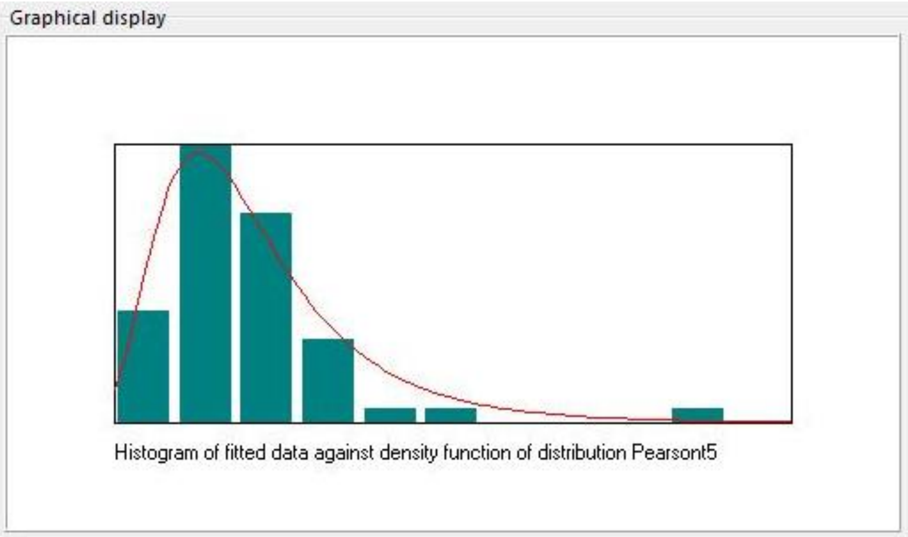
Figure 7 A boxplot of the storage time in days for scenario 1 for different runtimes and number of runs. 1: 365 days, 500 runs; 2: 182.5 days, 500 runs; 3: 365 days, 250 runs; 4: 250 days, 500 runs.

4.4 Current situation

For the simulation model, I started building the current situation of the food bank. I investigated what the different parts of the food bank are and I built the simulation model with the available buildings blocks in the simulation software. At the start of the model the donations are coming in. These donations go to the administration desk where they get registered. After registration the donations go to a storage location by a forklift driver. After the products are stored they get retrieved on some day by a forklift driver again and they end at the assembly line. The different blocks in the simulation software are called atoms.

4.4.1 Donations

The source atom is the atom that generates incoming donations. At first the inter arrival time of the donations is set. Based on the dataset from 2017, I determined the average incoming number of donations per week, without using the weekends. The weekends were not used, because during the weekend no donations are coming in. This resulted in 52 averages and these averages I divided by the number of seconds that are in 5 days which is 432000. By doing this I got the average inter arrival time per week and these averages are to calculated the distribution with ‘Autofit’. The averages per week instead of the single donations are used, because by using the single donations there was too much fluctuation between different periods in the year and it was not possible to calculate a distribution that fit well. In Figure 8 the distribution of the inter arrival is shown. With a score of 5 the Pearson 5 distribution seems to be a very good fit. This distribution has a mean of 4014.41, and an alpha parameter of 6.90. Which means that approximately 107 products are coming in each week. However, as



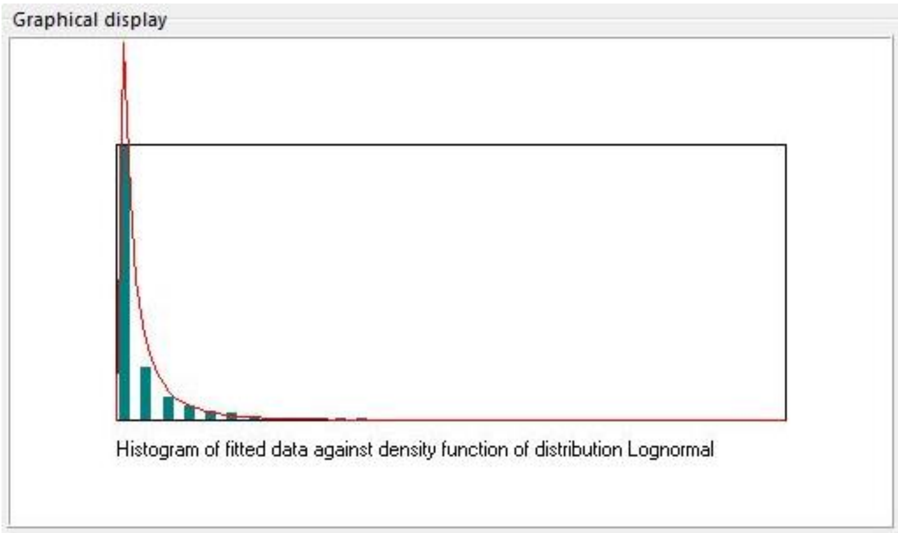
All Fitting Results

	Distribution	P1	P2	P3	Score
1	Pearsont5	4014.41	6.90	0.00	5
2	Lognormal	4014.41	1815.66	0.00	6
3	Erlang	4014.41	5.00	0.00	11
4	Gamma	4014.41	4.90	0.00	15

Figure 8 Output of the tool ‘Autofit’. Data from the average inter arrival time is used to calculate an appropriate distribution. The score indicates how good the distribution fits the dataset. For the Pearson 5 distribution the score is 5 which indicates that the distribution fits the data very well.

explained before, this distribution was not very reliable when the number of runs was tested and so it was adjusted. The new distribution is a uniform one with a lower boundary of 3927.273 and an upper boundary of 4547.368. This means that on average 102.5 products are coming in each week. In the first 24 hours a different distribution is used so that 200 products are coming in. This was done to create a buffer in the distribution centre, because in reality a distribution centre is never empty. It is not very usual to change an entire distribution to make a model more reliable. However, for this simulation model it does not influence the outcome of the model that much. The most important factor is that a steady state is reached in which the number of incoming and outgoing products is balanced. If this is not the case there could be, for example, more expired products, because the number of incoming products is higher than the number of outgoing products. With the first distribution a steady state was not reached and so it was adjusted not to influence the final results.

The second distribution of the simulation model that was defined was the expiration date of the incoming donations. First, based on the dataset from 2017 a distribution based on the shelf life per product was calculated. This resulted in a distribution that includes 6464 data points and it is shown in Figure 9. With a score of 6998 this distribution is not a very good fit of the data. After this, I calculated the average shelf life of the incoming donations per day from the day that the products were donated to get a distribution that fits better. I calculated the average shelf life per day of the products that were incoming on that day. This resulted in 365 data points of which some were 0, because as said before, during the weekend no products are coming in. I deleted all the points that were 0 from the list and with the remaining points the distribution was calculated. This resulted in a negative exponential distribution with a mean



All Fitting Results

	Distribution	P1	P2	P3	Score
1	Lognormal	16.13	27.60	0.00	6998
2	Negexp	16.13	0.00	0.00	7049
3	Weibull	16.13	0.70	0.00	7820

Figure 9 A distribution of the shelf life of the incoming products calculated for the shelf life per incoming product. The data contains 6464 data points. The score of this distribution is 6998 which implicates that the distribution is not a very good fit for the data.

of 8181654.61 and a score of 50 with the tool 'Autofit', so this distribution has a much better fit. In days, the mean of this distribution is approximately 94.

The third and last distribution I defined for the incoming donations is the number of products per donation. I calculated the average number of incoming products per day and, just as with the expiration date, this resulted in 365 data points. I deleted the ones that were 0 and 'Autofit' calculated the following distribution: a log-normal with a mean of 770.36, a standard deviation of 692.41, and a score of 33.

A side node for the parameters inter arrival time and expiration date is that these parameters are calculated in seconds. This is done, because the default setting in Enterprise Dynamics is seconds.

4.4.2 Administration

The second atom in the model is a queue in which the donations end up after they have been created at the source atom. This is a waiting area in which the donations wait until they get registered at the administration desk. I defined a label at this atom which indicate to which main product group the product belongs. Based on the dataset from 2017 there are nine main product groups. However, the groups that contain products that need to be stored in the fridge or freezer I deleted from the dataset so that leaves me with seven groups: eggs; bread; coffee and tea; groceries; grain and rice; potatoes, vegetables and fruit; and other. These groups occur in the dataset with a certain percentage. For example, 21.8% of the products belongs to the category groceries. I used a uniform distribution to draw a number between 0 and 1. If that number is smaller than the given percentage of the mean product group a product is assigned to that group. See for a more detailed description Appendix A and look at the atom description at the atom named 'ready for administration'.

4.4.3 Split large deliveries

The food bank in Rotterdam makes 2000 food crates per week. Based on a distribution as explained before the incoming donations get a number of products assigned. Sometimes these donations contain more than 2000 products. If this happens, the donation cannot be used in one week, because people get more of the same product in one week and often they do not want that. To avoid this, I inserted an atom that can split large incoming donations. So, after the splitting atom the number of products is split in half, but the expiration date stays the same for both new donations. By testing if this atom is doing its work I found out that there are a few pallets that contain even more than 2000 products after they are split in half. However, this happens only one or two times a year, so I assumed this as negligible.

4.4.4 Product storage

In the current situation of the food bank the products are stored in a random way. So, the forklift driver stores the products at some random place in the distribution centre. After this has happened the product stays in the distribution centre. This is dependent on when an order is generated that retrieves the product from the distribution centre. There are 14 different storage locations in the distribution centre as can be seen in Figure 4. For the locations further away from the assembly line it the transport takes longer so for each location I measured the distance in the distribution centre. I converted these distances to speed because in Enterprise Dynamics it is more convenient to work with speed instead of distance. To calculate the speed I determined the speed of a forklift at 5.5 km/h (Arbo, 2011; Dohmen, 2008). The time it takes for the forklift driver to drive from a given location in the distribution centre to the assembly line is the speed divided by the distance. However, in the model the distance of the fast non-accumulating conveyor is by default 10 meters. So, as a second step I divided this distance by the time it takes for the forklift driver to travel from the location to the assembly line and this gives me the final speed of the fast non-accumulating conveyor. After the products are retrieved from the shelves, they are delivered at the assembly line.

4.4.5 Order generation

Besides that donations are coming in, also food crates have to go out of the distribution centre. To simulate this an order generator was created. The first order is created after 13 days, so that the distribution centre has time to fill with some products. After that 13 days, the inter arrival time of orders is one week. I chose to use 13 days, because in the first 24 hours 200 products are entering the distribution centre. After these 200 products the distribution centre is filled with some products so that when an order is coming in there is no shortage of products. By running the model for two weeks more before the first order is coming in there is time to let products come in with the 'real' settings. This period could also have been a few days longer, but not too long because then the distribution centre cannot handle the number of products. The orders contain between 95 and 110 products. The order determines whether a product is retrieved from the shelf or not. After the order is collected, it is unpacked again. This is not what happens in the real situation, but with unpacking the order again the labels per product can be registered and measured. After the unpacking the products go to the assembly line.

For some products it is not a problem if they are handed out after the expiration date, but for some products this is a problem because of health issues. Based on the 'Nederlandse Voedsel- en Warenautoriteit (NVWA, n.d.) products that are expired go to a separate storage department which means that they are thrown away in reality. The products groups for which this applies are: eggs; bread; and potatoes, vegetables and fruit.

4.4.6 Time schedule and control availability

The food bank is not opened 24 hours a day and 7 days a week, so I designed a schedule on which hours the food bank is opened and on which not. I decided to open the food bank 5 days a week and on these 5 days the food bank is open for 24 hours. In real, the food bank distribution centre is opened around 6 hours a day 5 days a week. By adapting the inter arrival time the number of incoming products is more spread out over 5 days in the simulation model compared to the real situation in which the food bank is opened 6 hours a day. This has no consequence for the outcome of the model, because the inter arrival time is adapted in such a way that the number of incoming products per week is the same for the simulation model as for the real situation.

4.5 Class-based storage based on expiration date

As explained in section 3.2 there are many characteristics on which classes can be based. For this scenario I decided to base the classes on the expiration date of the incoming donations. Products that have a short expiration date are located in the same area; products that have a medium expiry date are located in the same area; and products with a long expiry date are located in the same area.

The main area in the model where things are changing compared to the current situation is the allocation of products. Based on the dataset from 2017, I made a division for the three different groups. All the products that have a shelf life of 30 days or shorter are sent to the most favourable locations in the distribution centre which is called group A. In group B are the products that have a shelf life between 30 and 70 days, and products that have a shelf life of 70 days or longer go to the least favourable locations in the distribution centre which is called group C.

Based on the distance from the assembly line the locations in the distribution centre are favourable, medium favourable, or not favourable at all. In Table 1 all the storage locations are ranked in the last column. In group A are: C2, C1, B1, and A1. Group B contains: D1, E1, B2, and F1. In group C are: G1, D2, A2, E2, F2, and G2. See Figure 4 again for an overview of the different storage locations. Appendix B contains the more detailed formulas that change by applying this scenario.

4.6 Class-based storage based on product categories

For this scenario I will use class-based storage as explained in section 3.23.3. In this way of storage I will look at the different product groups and if these groups can have an effect on the efficiency of the distribution centre. At the food bank the different main food groups are already registered and these groups are used for class-based storage. Based on the amount of products in the dataset of 2017 the locations are assigned; this results in seven classes. So, the product group with the most products in it is located at the most favourable location in the distribution centre. The most favourable locations are the same as for the class-based storage. Within these groups it is possible to sort the products on expiration date. Also for this scenario the only place in the model where things are changed is at the allocation of products. The detailed formulas are shown in Appendix B.

5 Results and discussion

In this chapter the results of the scenarios described in the previous chapter are given. For every performance measure first the results and statistical analysis are given and second a discussion about the results is given. The performance measures that were measured are: the number of expired products, the time spent on the storage location, and the travel time in the distribution centre. For the analysis of the results SPSS® version 23 release 23.0.0.2 was used. A one-way ANOVA test was used for all performance measures to test whether a difference between the different groups exist and if a difference exists a post-hoc test was used to see which specific groups differ from each other. The significance level that is used for all the statistical tests is $\alpha = 0.05$.

5.1 Expired products

The performance measure expired products counts the number of products that are expired when they are leaving the simulation model, or, in reality, when they are collected for the food crates. With products the pallets with products on it is meant. At the start of the model products receive and expiration date and when this expiration date is reached before they leave the model the product counts as expired. As explained before, there are three categories of products which are not dangerous for human health when expired. These products are still used for the food crates and do not count as expired.

After running the model the number of expired products is compared between the different scenarios. A boxplot of this comparison is made which is shown in Figure 10. A one-

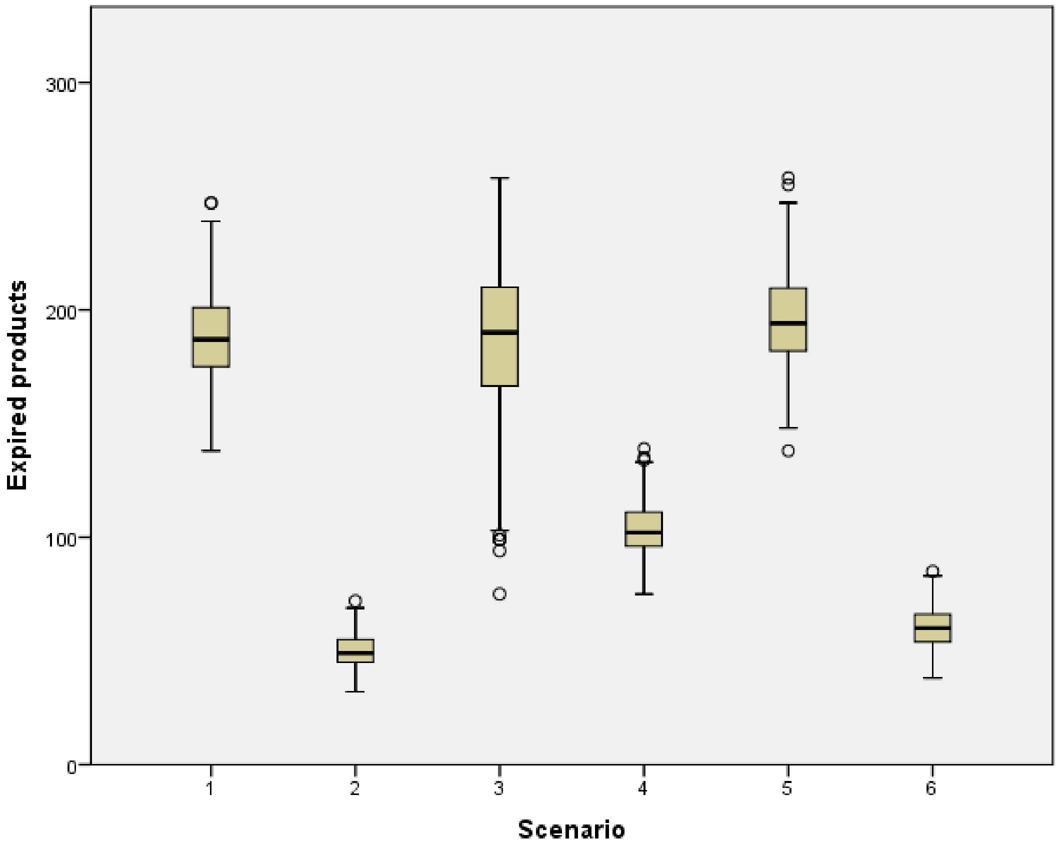


Figure 10 A boxplot of the number of expired products given per scenario. The circles are the outliers of the dataset.

way ANOVA test is done to see if there is any difference between the scenarios. With a P-value of 0.000 it is shown that there is a difference between the scenarios. To see where exactly this difference occurs a post-hoc test is performed. All scenarios are significantly different with a P-value of 0.000 except for scenario 1 and 3; they are not significantly different from each other with a P-value of 0.972. Although it looks like scenario 1 and 5 are also not significantly different from each other they are significantly different with a P-value of 0.000.

The average number of incoming pallets with products is 3670 per run. As a result of these tests it can be said that scenario 2 has the least expired products with a mean of 49.79 which is 1.36% of the total number of incoming products followed by scenario 6 and scenario 4 with a mean of 60.01 and 103.52 respectively which are 1.64% and 2.82% of the total number of incoming products. Scenario 1, 3, and 5 have a lot of expired products with a mean of 187.74, 185.60, and 195.55 respectively which is 5.12%, 5.06%, and 5.33%of the total number of incoming products.

This can be due to the fact that in these scenarios products are not sorted on expiration date. Every new incoming product is placed at the end of the queue. For products with a short expiration date this is not very convenient, because they are often expired when leaving their storage location. Compared to scenario 2, 4, and 6 sorting products has an impact on the number of expired products; the mean of scenario 2 is 73.4% lower compared to scenario 1. Within the two different ways of sorting the products there is also some difference between the scenarios. Scenario 2, 4, and 6 are all sorting products based on expiration date, but scenario 4 has more expired products compared to scenario 2 and 6. This can be a result of the assignment policy of the different scenarios. For scenario 2 products are randomly stored in the distribution centre and they are also evenly spread throughout the distribution centre. This results in queues that have all approximately the same length. For scenario 4 there are 3 classes and each class uses their best storage location first. As long as it is not completely filled it does not use the second best location. This results in long queues in some storage locations and no products in other locations; products are not evenly spread throughout the distribution centre. When products with a short expiration date enter a long queue they will be put somewhere in the front, but there are more products with a short expiration date in a long queue so it might happen that they still have to wait for a long time while in many short queues products with a short expiration date are evenly spread throughout the distribution centre and they are picked sooner. Scenario 6 is in between scenario 2 and 4 and this scenario has 7 classes. In randomized storage as in scenario 2 it can be said that every location in the distribution centre is a class so it contains 14 classes. As a result it can be said that the more classes a storage assignment policy contains the less products are expired. However, this is only applicable if products are sorted on expiration date while stored, because the effect is not shown in scenario 1, 3, and 5.

5.2 Time spent on storage location

For this performance measure the time that a product is stored is measured. It is measured from the time the product arrives at the storage location till the time the product is retrieved. From the outcome of the simulation study a boxplot is made which is shown in Figure 11. It can immediately be seen that scenario 2, 4, and 6 have a lot of outliers and extremes. With

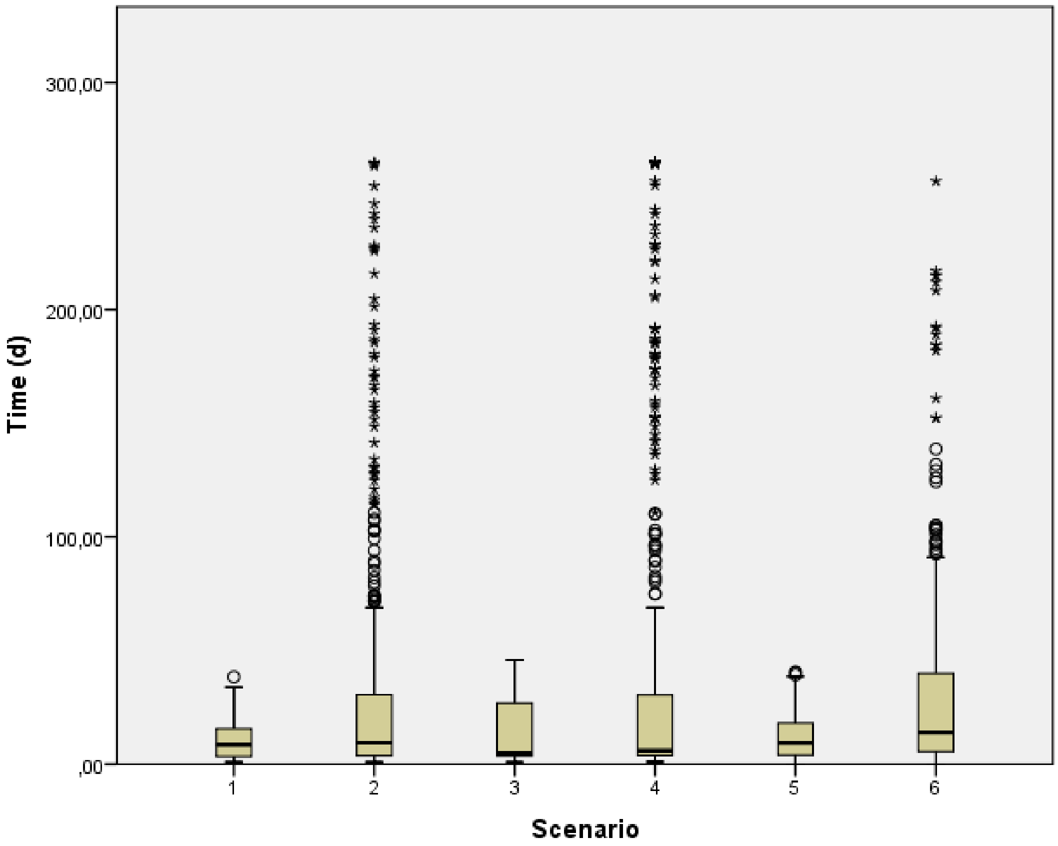


Figure 11 A boxplot of the time a product stays on the assigned storage location given per scenario. The circles and stars are the outliers and extremes of the dataset.

and one-way ANOVA test it was shown that there is some difference between the scenarios, because the P-value of the test is 0.000. With a post-hoc test the differences are shown in more detail. Scenario 1 and 5 are not different with a P-value of 0.323. Scenario 2 and 4, and 2 and 6 are also not different with a P-value of 0.808 and 1.000 respectively. Scenario 3 and 5 are not different with a P-value of 0.062, and scenario 4 and 6 are not different with a P-value of 0.229.

The mean time a product stays on a storage location for scenario 1, 3, and 5 is 10.15, 13.49, and 11.35 days respectively. For scenario 2, 4, and 6 this is 32.51, 38.89, and 30.45 days respectively.

The differences between the scenarios are all based on the outgoing policy of the scenarios. Scenario 1, 3, and 5 do not sort products based on expiration date and scenario 2, 4, and 6 do sort products based on expiration date. The outcome of the simulation model is actually quite logical. If in scenario 1, 3, and 5 a product is coming in with a short expiration date is it placed at the end of the queue. Every product stays approximately the same time in the distribution centre, because each week approximately the same amount of products is needed for the crates. For scenario 2, 4, and 6 this is different. If a product with a short expiration date is coming in it is placed in front of the queue so that it does not get expired. If a product has a long expiration date it will stay longer on its storage location in scenario 2, 4, and 6, because products with a shorter date go in front of the queue. The result of 'first expired first out' policy is that, on average, products stay longer on the storage locations.

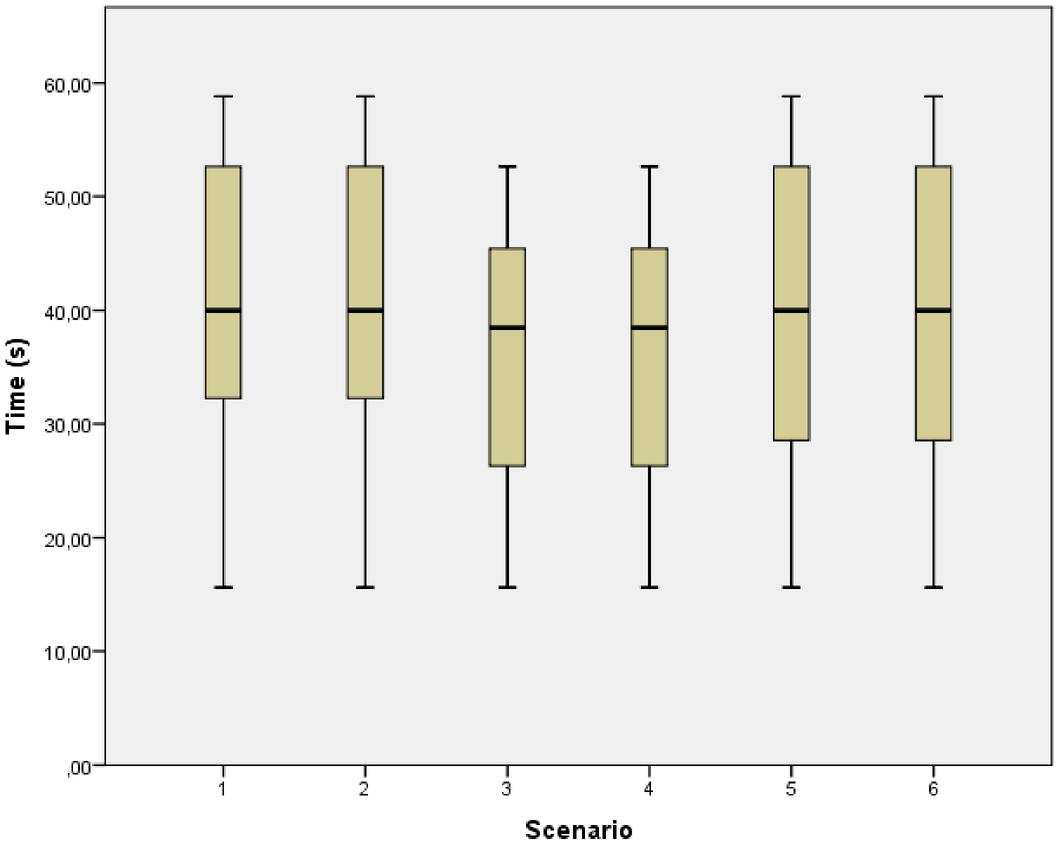


Figure 12 A boxplot of the time in seconds that it takes to store and retrieve products given per scenario.

5.3 Travel time

The travel time of the products is the time it takes to bring the products to their assigned storage location and to retrieve them and bring them to the assembly line. The distance from the assembly line to a given storage location is different for each location, so the travel time is also different. There are three different storage assignment policies used and between these policies a difference is expected, but within these policies you would expect the same travel time. In Figure 12 a boxplot of the travel time in seconds per scenario is given. At first sight the three groups of scenarios look different from each other, but the same for each storage assignment policy. With a one-way ANOVA test it was tested if there is any difference between the mean of the scenarios. The outcome of this test gives a P-value of 0.000 which means that there is a difference between the scenarios found. To find the exact difference a post-hoc test was performed. The outcome of this test is that scenario 1 and 2 are not different from each other just like scenario 3 and 4 and scenario 5 and 6; this is an outcome I expected in advance. A more unexpected outcome of this test is that scenario 2 and 5 are not significantly different with a P-value of 0.006.

The mean travel time of scenario 1 and 2 is 41.04 and 40.25 seconds respectively, the mean travel time of scenario 3 and 4 is 34.83 and 34.71 seconds respectively and for scenario 5 and 6 the mean travel times are 38.94 and 38.64 seconds respectively. As a result it can be said that the travel time in a distribution centre is reduced if class-based storage with a low number of classes based on expiration date is used. However, the travel time used in this model is calculated in seconds so it is questionable how big the impact of this reduced travel time really is. For example, if 6464 products are coming in in one year and scenario 2 is used the total travel time is 72.27 hours. If 6464 products are coming in and scenario 4 is used the total travel time is 62.32 hours. This is a reduction of 9.95 hours on annual basis which is 13.8%, so this is approximately only a bit more than one working day.

5.4 Class-based storage time

The class-based storage assignment policy sends products to storage locations based on their expiration date. The reason to do so is that products that have a short expiration date are used soon so they are stored at the most desirable storage locations and as a consequence the walking or driving distance is reduced. However, if a product has a short expiration date it is not guaranteed that it is actually used first as well. To test if this is the case the time a product stays on the storage location is split up per class for scenario 3; the boxplot of this is shown in Figure 13. The mean storage time for class A is 14.41 days, for class B this is 21.96, and the storage time for class C is 27.91. A one-way ANOVA was performed and with a P-value of 0.000 it is shown that the groups are significant different; a post-hoc test gives that all the groups are significant different with a P-value of 0.000. This means that if a product has a short expiration date (class A) it will on average stay shorter in storage than a product that has a long expiration date (class C). This comparison between the classes is only done for scenario

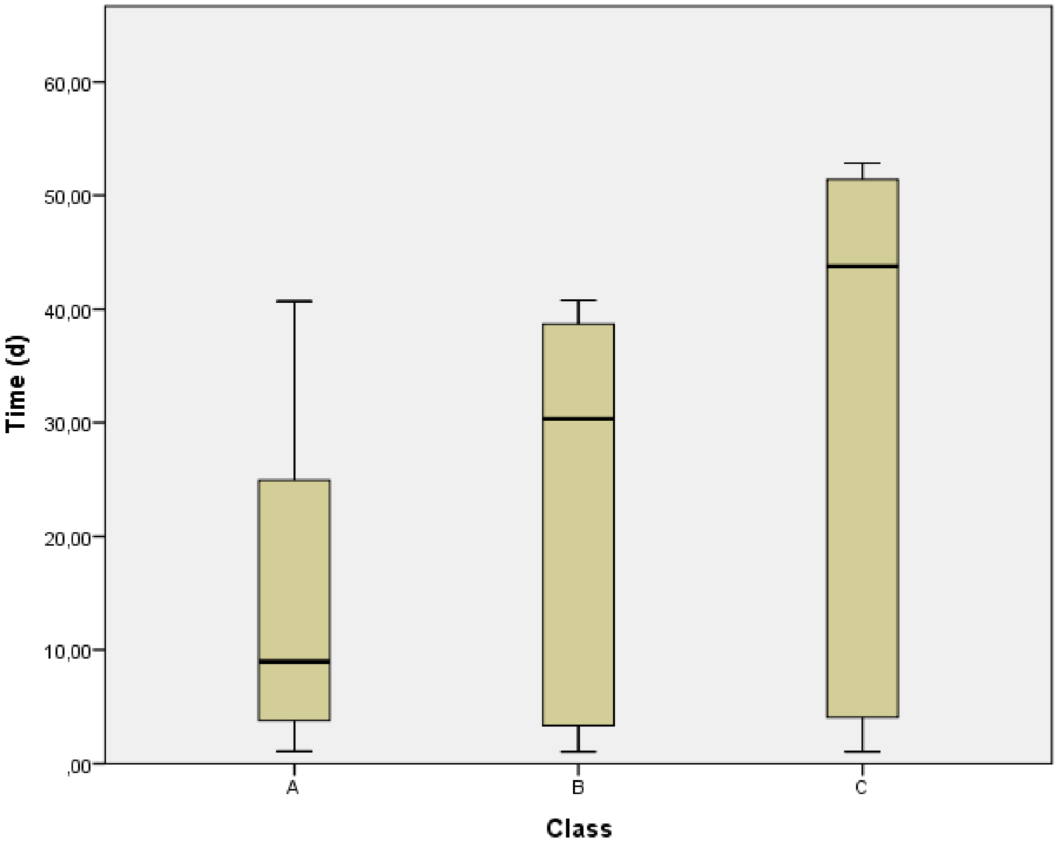


Figure 13 A boxplot of the average time in days that a product stays on the storage location for scenario 3. The classes are based on the expiration dates of the products. Class A contains product which have an expiration date of 30 days or less, for class B the boundaries are 30-70 days, and Class C contains products with an expiration date longer than 70 days.

3, because in scenario 3 products are assigned to storage locations based on classes. This also applies to scenario 4, but it works the same as for scenario 3.

5.5 Sensitivity analysis

To test the robustness of the model a sensitivity analysis was done. By slightly adjusting some parameters the robustness of the model is tested. If the outcome of the model is the same when the parameters are changed the model is robust, but when the outcome of the adjusted model is very different compared to the default model, the model is not very robust (Aalst, 1995).

For the first sensitivity analysis scenario 4 is the default. This scenario is chosen, because for this scenario the storage assignment policy as well as the outgoing policy are related to the expiration date of the products. So, I would expect the largest changes in the number of expired products for this one. The distribution of the expiration date is adjusted with different percentages to see what the effect on the number of expired products is. As a consequence, also the boundaries of the classes need to be adjusted with the same percentage as the distribution. There are six extra scenarios run and the results of these runs are shown in Figure 14. If the expiration date is shortened by 30% the mean number of expired products is 140.51 which is 35.7% more compared to the default scenario which has a mean of 103.52. If the expiration date is extended by 30% the mean number of expired products is 83.22 which is 19.6% lower compared with the default scenario. So, if the expiration date becomes shorter the number of products that will expire increases faster compared to if the

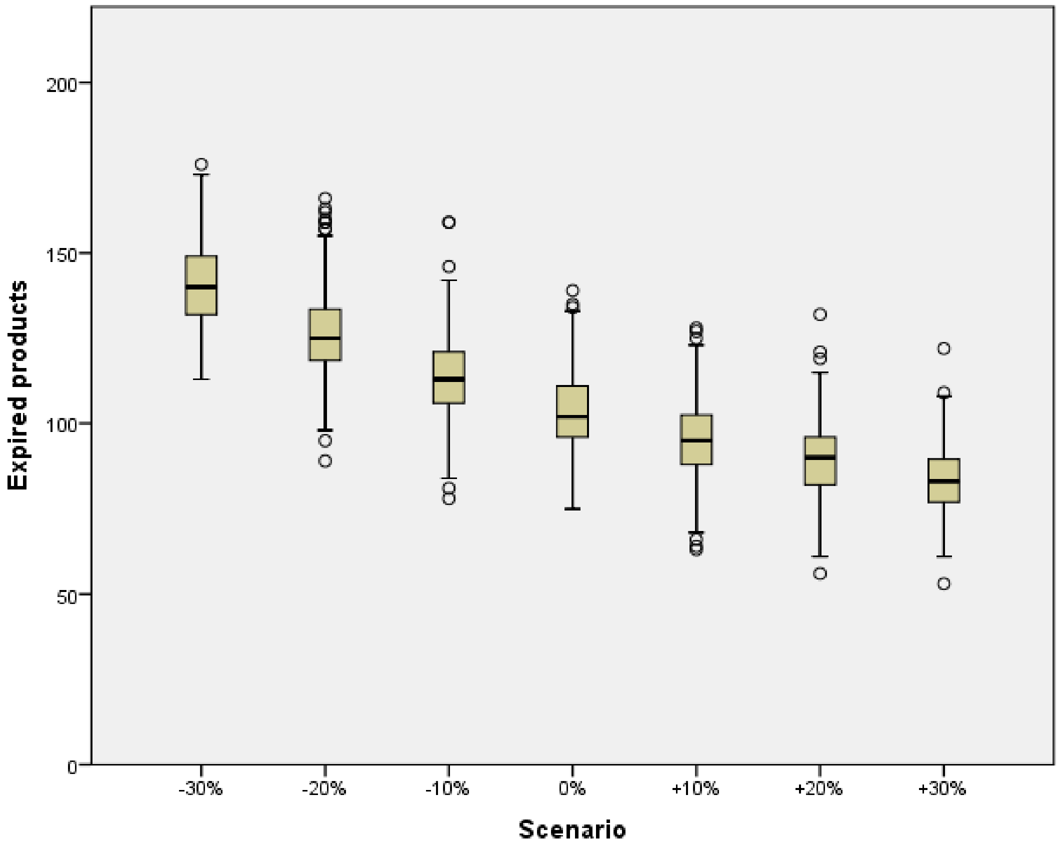


Figure 14 A boxplot of the number of expired products per scenario compared with main scenario 4.

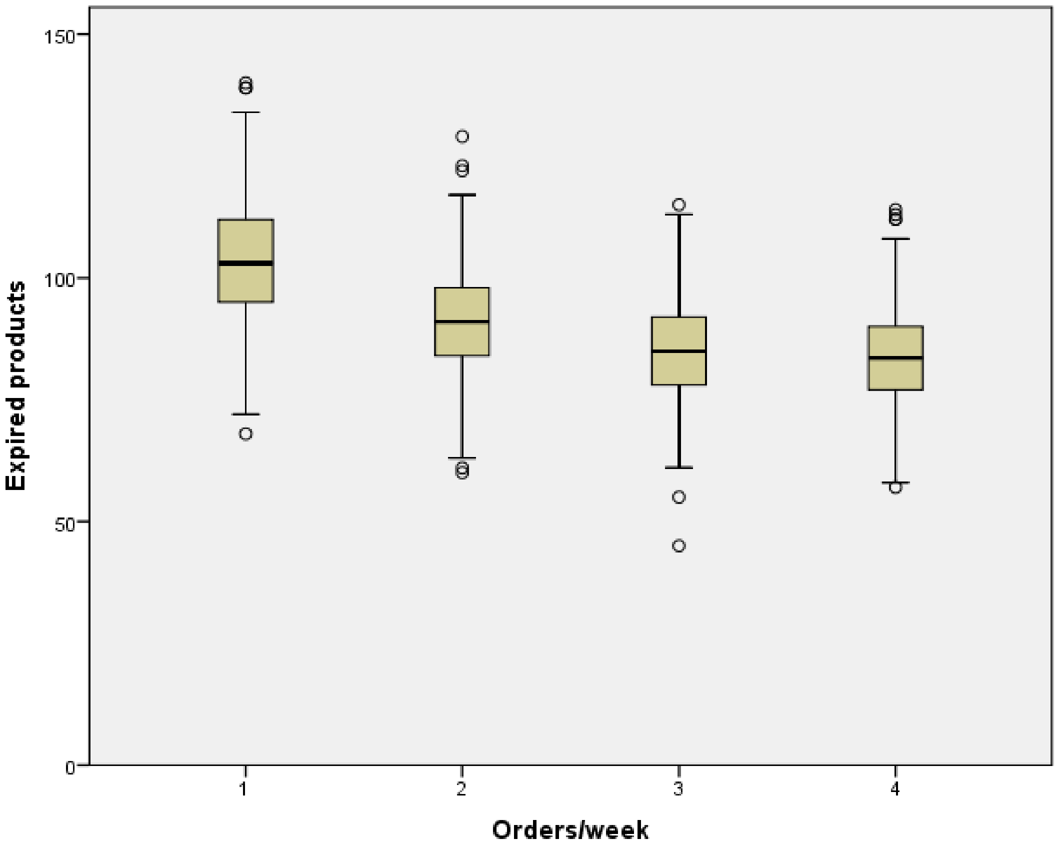


Figure 15 A boxplot of the number of expired products for a different number of orders per week. Scenario 4 is used as default.

expiration date becomes longer. If you would draw a line through the boxplots in Figure 14 the left half has a higher slope compared to the right half of the figure.

For a second sensitivity analysis a distribution that can be adjusted in the model is the number of orders that is generated. In the model there is one order generated every week, but if there are more order points in one week, what happens in reality, it might be that the number of expired products is less. To test this, scenario 4 is used as default again. The inter arrival time of the orders is adjusted from 2 per week to 4 per week and the number of products an order contains is also adjusted so that the mean number of used products per week stays the same. The results of these runs are shown in Figure 15. As with the previous sensitivity analysis a decreasing line can be drawn through the boxplots. With a one-way ANOVA and a post-hoc test it was tested if there are significant differences between the scenarios. Only scenario 3 and 4 are not significantly different with a P-value of 0.134. The conclusion of this analysis is that if there are more order points during a week the number of expired products is decreasing. However, at some point it does not matter anymore if the number of order points is increased, because the number of expired products stays approximately the same.

6 Conclusion and recommendations

The main question of this research is: What is the optimal design of the internal logistics regarding the assignment of storage locations of the food bank distribution centre in Rotterdam? In this chapter the answer to this question is given. Also the assumptions that are made for the simulation model are discussed, future research recommendations are made and a comparison to other previous studies is done.

6.1 Conclusion and discussion

In the scenarios with the 'first expired first out' outgoing policy the number of expired products is less compared to scenarios with the 'first in first out' outgoing policy. So, sorting products on their expiration date leads to less expired products and as a consequence less food waste. For the storage time of products the scenarios with the 'first in first out' outgoing policy have shorter storage times compared to the scenarios with the 'first expired first out' outgoing policy. It can be said that if products are not sorted on their expiration date they will stay shorter in the distribution centre. The last performance measure, the travel time, is shortest in the two scenarios with the class-based storage assignment policy based on the expiration date of the products.

Based on the results the following conclusions and recommendations can be made on choosing which is the best applicable scenario for the food bank in Rotterdam. Also the preference of the decision maker must be taken into account. This means that for a decision maker certain performance measure can be of more importance than other performance measures. So, if the decision maker thinks the storage time of products is the most important performance measure, then the outcome of this thesis is different compared to choosing for the number of expired products as the most important performance measure. For this conclusion the number of expired products and the travel time in the distribution centre are chosen as the most important performance measures. With this in mind class-based storage based on expiration date as storage assignment policy and a 'first expired first out' outgoing policy (scenario 4) is the best improvement of the current scenario which has randomized storage as storage assignment policy and a 'first in first out' outgoing policy (scenario 1). Compared to the current situation the mean number of expired products is reduced with 44.9% and the mean travel time in the distribution centre is reduced with 15.4% for the new scenario. For the last performance measure, the storage time of the products, the mean time of the new scenario increased with 283.2% compared to the current scenario.

In general it can be said that with this thesis the use of class-based is preferred over the use of randomized storage. This is the case if products are sorted based on their expiration date and if the travel time is of major importance and if the throughput time is of minor importance.

However, after the pervious described results one step is missing and this is the implementation of the results. As said before, the food bank is an organization existing of volunteers only. This makes it hard to implement new scenarios, because no one is working fulltime. Also the level of education is very different between the volunteers. With the simulation model that is made in this research there are some improvements possible, but the most important thing is that these improvements are implemented in a good way so that it is

Table 3 A possible way of how a poster could look that will help the volunteers of the food bank by deciding to which storage location in the distribution centre they have to send a pallet.

Expiration date	**Location**					
Importance	*1*	*2*	*3*	*4*	*5*	*6*
Less than 30 days	C2	C1	B1	A1		
Between 30 and 70 days	D1	E1	B2	F1		
More than 70 days	G1	D2	A2	E2	F2	G2

understandable and usable for everyone. Someone more in the field of communication could take a look to the outcomes of this research and think of a way to implement them. An example of a solution could be a poster on which the three different classes of expiration dates are given. Also the storage locations in order of importance are given so that it is clear for everyone where a pallet needs to be stored in the distribution centre. A very basic example of this poster is given in Table 3.

6.2 Assumptions

In a simulation model there are always assumptions that have to be made. In this paragraph these assumptions are discussed.

When products enter the model they get a label with the number of products a pallet contains. When the crates are filled with food products I wanted to use the number of products on a pallet to determine the number of pallets that was needed to fill a crate. So if a pallet contains a lot of products, less pallets are needed than when a pallet contains little products. However, this was very difficult to simulate in Enterprise Dynamics, so I assumed that every pallet contains the same amount of products and with a distribution the number of needed pallets is eventually varied between 95 and 110. This might result in a different number of pallets that is taken for the food crates compared to the real situation and this might have an effect of the number of expired products for example, but I think this effect is not really big. A second assumption that has a larger influence on the outcome of this research is that when picking the orders product categories are not taken into account. A food crate needs to be diverse; it should contain products from every product category. The simulation model I made is only taking into account the number of pallets that is needed and not the product category. Firstly, this could lead to a crate that is only filled with grain products and that is not a very good reflection of reality. Secondly, if products from different categories are taken this can have an effect on the number of expired products, because if from one category a lot of products are present less products are needed relatively. This results in a lot products from a category that are left for next week of the week after that and this increases the chance of products getting expired.

In scenario 2, 4, and 6 the outgoing policy of the products is 'first expired first out'. The product with the shortest date is put in front of the queue. In an ideal situation the products are sorted based on all the different storage locations. In the simulation model the products are only sorted within their specific storage location. There are 14 storage locations so 14 different queues in which products are sorted. It can still happen that if products are taken from location A1 the products from location B2 have a shorter expiration date. I chose to do it this way, because it was very difficult, and for me impossible, to model it the way I wanted to model it. I think this has a large effect on the number of expired products and also on the travel time, because now not always the products with the shortest expiration date are taken. I think the travel time can be further reduced by implementing selection on expiration date for all the storage locations, because the least preferable locations will then be chosen even less often compared to the current simulation model and this will decrease the travel time.

The delivery schedule of the food bank in the simulation model is that it is opened 5 days a week 24 hours per day. The other two days incoming donations are not allowed, but in these two days the order for the food crates is coming in. In the real situation of the food bank the donations are coming in 5 days a week, but not 24 hours a day. I think this does not have a major influence on the outcome of this thesis, because the inter arrival time is adjusted in such a way that the number of incoming products per week is the same in the current simulation model compared to the real situation.

The number of orders that is coming in is different in the model from the real situation. In the model an order is coming in once a week while in reality the food bank is making crates four times a week. As explained in section 5.5 this reduce in order points has an influence on the number of expired products at the food bank. However, because the number of order points is the same for all the scenarios the influence is of minor importance.

6.3 Future research

In the previous section the assumptions are discussed. To make to model more feasible one could look at the assumptions again and try to think of solutions for them so that the model becomes more realistic. The assumption that can be improved most is the one about the sorting based on expiration date. Now the products are sorted within their storage location, but the reliability of the model could be improved by sorting all the products together on their expiration date. A result of doing this can be that the products with a short expiration date are chosen more often than that is the case now. And if the products with the shortest expiration date are chosen more often this means that the most preferable locations are used more often which reduces the travel time in the distribution centre. Also the compiling of the food crates could be of an influence when changed in the simulation model. It would probably make a difference when there is looked at the ingredients of the crates. If from every product category one or more products are taken a varied food crate is formed. This in combination with looking at the sorting on expiration date for the model as a whole will improve the model and make it more realistic.

In another study about the implementation of class-based storage one of the results was that the choice of how storage classes are assigned to storage locations has a major impact on the successful implementation of class-based storage (Petersen et al., 2004). In the research for the food bank I did not take a look at the assignment of classes to storage locations. I only determined the distance from each location to the assembly line to identify what the most favourable locations in the distribution centre are. The implementation of class-based storage could be further improved when looking at what the effect of assigning classes to storage locations is.

In Figure 5 the main processes and decisions of the food bank are shown and also the scope of this thesis is shown; the processes and decisions that are included in the simulation model. For future research the current simulation model could be extended so that more processes and decisions are included. If, for example, the process of which donations to accept and which not is included this could have an effect on the number of expired products if products with a short expiration date are no longer accepted. However, it is questionable if this is feasible for the food bank, because there are not always enough products and they have not always something to choose. A second process that might improve the current simulation model is including the chance of the storage location of the products being reported wrong to the administration desk. When this happens it is not easy to find a lost pallet back and it can be forgotten. If this process is included a logical design of the distribution centre becomes more important compared to this thesis. This means that, for example, if a distribution centre is divided in small areas or classes it is easier to find a pallet back if you know that it has expiration date of less than 30 days, because there is only a small area of the distribution centre in which products with an expiration date of less than 30 days are stored.

Another process that might improve the current simulation model is the process described in section 2.4. This is the process of dividing the incoming products already for different location in Rotterdam or the Netherlands. Including this process might have an effect on the recruitment department that have to decide which products to accept and which not. If, for example, the food bank in Rotterdam decides to no longer accept donations that are meant for the rest of the Netherlands they can reduce the storage space of the distribution centre and the transportation costs as well.

6.4 Comparison with other studies

A study in which the storage assignment policy for a food bank is determined is not done before. However, a lot of research is done in which the storage assignment policies are compared or applied to different situations.

In a study of Zhang (2016) correlated storage is used to reduce the travel distance in a picker-to-parts order picking system. The conclusion is that to reduce the average travel distance per picking, correlated stock-keeping-units should be stored near each other in the warehouse. However, the picking frequency plays the most important role in storage location assignment. He says that these two factors together need to be considered for the development of algorithms.

These results are quite similar to the ones found in this thesis. In this thesis the items in the class-based storage scenario are not statistically correlated, but by storing products based on their expiration date the average travel distance per picking is reduced. Also the picking frequency, or in this thesis the order frequency, is at the sensitivity analysis mentioned as something that can reduce the number of expired products. In future research an algorithm could be developed to combine these two factors for the food bank.

In another study about improving the productivity of a distribution centre through implementation of class-based storage one of the results is that the use of a storage assignment policy is greatly dependent on the type of the storage system. Also the use of combination of factors and the selection of performance indicators is important, because they can react differently under different storage policies (Chan & Chan, 2011). For the study of the food bank the type of storage system is not specifically taken into account. However, the choice of the performance indicators has an influence on the outcome of the results. At first, the performance indicators react differently under different scenarios. Secondly, it depends on the preferences of the decision maker which scenario is preferred over the others, because there is not one scenario in which all the performance indicators are the best; there is a trade-off that has to be made.

References

Aalst, W. M. P. van der. (1995). *Handboek simulatie*. Eindhoven.

Arbo. (2011). Het risico: Heftrucks. Retrieved June 22, 2018, from http://www.arbo-online.nl/rie/artikel/2011/06/het-risico-heftrucks-2-10110065

Bangsow, S. (2012). *Use cases of discrete event simulation: Appliance and research*. Springer Berlin Heidelberg.

Bindi, F., Manzini, R., Pareschi, A., & Regattieri, A. (2009). *Similarity-based storage allocation rules in an order picking system: An application to the food service industry*. *International Journal of Logistics Research and Applications* (Vol. 12).

Chan, F. T. S., & Chan, H. K. (2011). Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. *Expert Systems with Applications*, *38*(3), 2686–2700.

Ciervo, B. (2017). What information do I need to design my warehouse? Retrieved February 26, 2018, from https://www.conveyco.com/information-need-design-warehouse/

Davis, L. B., Sengul, I., Ivy, J. S., Brock, L. G., & Miles, L. (2014). Scheduling food bank collections and deliveries to ensure food safety and improve access. *Socio-Economic Planning Sciences*, *48*(3), 175–188.

de Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, *182*(2), 481–501.

de Ruijter, H. (2007). *Improved storage in a book warehouse*. University of Twente. Retrieved from http://essay.utwente.nl/58067/1/scriptie_H_de_Ruijter.pdf

Dohmen, M. (2008). Snelheid heftruck afgestemd op omgeving. Retrieved April 25, 2018, from http://www.logistiek.nl/warehousing/artikel/2008/06/snelheid-heftruck-afgestemd-op-omgeving-1019551

Ghanmi, A. (2006). Modeling and analysis of a Canadian Forces Geomatics division workflow. *European Journal of Operational Research*, *170*(3), 1001–1016.

González-Torre, P. L., & Coque, J. (2016). How is a food bank managed? Different profiles in Spain. *Agriculture and Human Values*, *33*(1), 89–100.

Gu, J., Goetschalckx, M., & McGinnis, L. F. (2010). Research on warehouse design and performance evaluation: A comprehensive review. *European Journal of Operational Research*, *203*(3), 539–549.

Jiang, S., Davis, L., Mleo, H. T. de, & Terry, J. (2011). Using Data Mining to Analyze Donation Data for a Local Food Bank.

Larson, P. D., & McLachlin, R. (2011). Supply chain integration under chaotic conditions: Not-for-profit food distribution. *International Journal of Procurement Management*, *4*(3), 315–322.

Malmborg, C. J., & Bhaskaran, K. (1990). A revised proof of optimality for the cube-per-order index rule for stored item location. *Applied Mathematical Modelling*, *14*(2), 87–95.

Ng, W. L. (2007). A simple classifier for multiple criteria ABC analysis. *European Journal of Operational Research*, *177*(1), 344–353.

NVWA. (n.d.). Houdbaarheidsdatum levensmiddelen. Retrieved April 25, 2018, from https://www.nvwa.nl/onderwerpen/etikettering-van-levensmiddelen/houdbaarheidsdatum-levensmiddelen

Parfitt, J., Barthel, M., & MacNaughton, S. (2010). Food waste within food supply chains: Quantification and potential for change to 2050. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1554), 3065–3081.

Petersen, C. G., Aase, G. R., & Heiser, D. R. (2004). Improving order-picking performance through the implementation of class-based storage. *International Journal of Physical Distribution & Logistics Management*, *34*(7), 534–544.

Rambeloson, Z. J., Darmon, N., & Ferguson, E. L. (2008). Linear programming can help identify practical solutions to improve the nutritional quality of food aid. *Public Health Nutrition*, *11*(4), 395–404.

Roodbergen, K. J., & Vis, I. F. A. (2009). A survey of literature on automated storage and retrieval systems. *European Journal of Operational Research*, *194*(2), 343–362.

Rouwenhorst, B., Reuter, B., Stockrahm, V., van Houtum, G. J., Mantel, R. J., & Zijm, W. H. M. (2000). Warehouse design and control: Framework and literature review. *European Journal of Operational Research*, *122*(3), 515–533.

Tako, A. A., & Robinson, S. (2010). Model development in discrete-event simulation and system dynamics: An empirical study of expert modellers. *European Journal of Operational Research*, *207*(2), 784–794.

van den Berg, J. P. (1999). A literature survey on planning and control of warehousing systems. *IIE Transactions*, *1999*(31), 751–762.

van den Berg, J. P., & Zijm, W. H. M. (1999). Models for warehouse management: Classification and examples. *International Journal of Production Economics*, *59*, 519–528.

van der Horst, H., Pascucci, S., & Bol, W. (2014). The “dark side” of food banks? Exploring emotional responses of food bank receivers in the Netherlands. *British Food Journal*, *116*(9), 1506–1520.

van Kampen, T. J., Akkerman, R., & van Donk, D. P. (2012). SKU classification: a literature review and conceptual framework. *International Journal of Operations & Production Management*, *32*(7), 850–876.

Wäscher, G. (2004). Order Picking: A Survey of Planning Problems and Methods. In *Supply Chain Management and Reverse Logistics* (pp. 323–347). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zhang, Y. (2016). Correlated Storage Assignment Strategy to reduce Travel Distance in Order Picking. In *7th IFAC Conference on Management and Control of Production and Logistics* (Vol. 49, pp. 30–35). Bremen.

Appendices

A. Model formulas for all scenarios

Table 4 Formulas in the simulation model, only the ones that are not default are given. If an atom name is printed in italic this means the information differs per scenario and it is explained in Table 5 . If the word 'parameter' is given, look in Appendix C for the exact number.

Atom name (characteristic)	Where in atom	Code
Donations (source)	Inter-arrival time [s]	`If(Time<Hr(24), 432, uniform(3927.273, 4547.368))`
	Trigger on exit	`Do(Label([THT], i):= Negexp(8181654.61), Label([Prod on pallet], i):= round(Normal(622.09, 198.41)))`
Ready for administration (queue)	Capacity	`1000000`
	Trigger on entry	`do(Label([distr], c):= Uniform(0, 1), if(Label([distr], c)<Parameter(15, 1), do(Label([product], i):= Parameter(15, 3)), if(Label([distr], c)<Parameter(16, 1), do(Label([product], i):= Parameter(16, 3)), if(Label([distr], c)<Parameter(17, 1), do(Label([product], i):= Parameter(17, 3)), if(Label([distr], c)<Parameter(18, 1), do(Label([product], i):= Parameter(18, 3)), if(Label([distr], c)<Parameter(19, 1), do(Label([product], i):= Parameter(19, 3)), if(Label([distr], c)<Parameter(20, 1), do(Label([product], i):= Parameter(20, 3)), if(Label([distr], c)<Parameter(21, 1), do(Label([product], i):= Parameter(21, 3))))))))))`
Administration desk (server)	Cycle time [s]	`Uniform(Parameter(28,1),Parameter(30,1))`
	Send to	`if(Label([Prod op pallet], first(c)) > 2000, 2, 1)`
Ready for forklift driver (queue)	Capacity	`1000000`

<table>
<tr><td></td><td>Send to</td><td><code>4. A random open channel: choose a random channel from all the open output channels.</code></td></tr>
<tr><td>Forklift driver 1 + 2 (server)</td><td>Cycle time [s]</td><td><code>Uniform(Parameter(27,1),Parameter(29,1))</code></td></tr>
<tr><td>Split large deliveries (splitter)</td><td>Naming rule</td><td><code>Add channel no. to name</code></td></tr>
<tr><td></td><td>Trigger on entry</td><td><code>Label([Big donation], i):= 1</code></td></tr>
<tr><td>Lb first product (server)</td><td>Cycle time [s]</td><td><code>0</code></td></tr>
<tr><td></td><td>Trigger on entry</td><td><code>Label([Prod on pallet], i):= round(0.5*Label([Prod on pallet], i))</code></td></tr>
<tr><td>Lb second product (server)</td><td>Cycle time [s]</td><td><code>0</code></td></tr>
<tr><td></td><td>Trigger on entry</td><td><code>Label([Prod on pallet], i):= round(0.5*Label([Prod on pallet], i))</code></td></tr>
<tr><td>Ready for storage place (queue)</td><td>Capacity</td><td><code>1000000</code></td></tr>
<tr><td></td><td>Send to</td><td><code>Look at</code> Table 5</td></tr>
<tr><td></td><td>Trigger on entry</td><td><code>Look at</code> Table 5</td></tr>
<tr><td></td><td>Trigger on exit</td><td><code>label([Entry time], i):= Time</code></td></tr>
<tr><td>Time schedule (switch availability)</td><td>Table of Time schedule</td><td><table><tr><td></td><td>Time</td><td>Down=1</td></tr><tr><td>1</td><td>Hr(0)</td><td>0</td></tr><tr><td>2</td><td>Hr(120)</td><td>1</td></tr><tr><td>3</td><td>Hr(168)</td><td>0</td></tr></table></td></tr>
<tr><td>Distance travelled A1 till G2 (fast non accumulating conveyor)</td><td>Capacity</td><td><code>10000</code></td></tr>
<tr><td></td><td>Speed [m/s]</td><td><code>Look at</code> Table 8</td></tr>
<tr><td></td><td>Trigger on exit</td><td><code>label([Distance],i):= Time-label([Entry time],i)</code></td></tr>
</table>

A1 till G2 (queue)	*Capacity*	*Look at* Table 7
	Queue discipline	*Look at* Table 5
Selecting products (server)	Input strategy	`5. Round robin with content check`
	Send to	`if(Label([Distance], first(c)) > 100, 2, 1)`
	Trigger on entry	`do(Label([WH time], i):= Time-Label([Entry time], i), Label([WH time], c):= Label([WH time], i))`
	Trigger on exit	`If(and(label([THT], i)<Label([WH time], i), label([product], i)=1), label([Expired], i):= 1, If(and(label([THT], i)<Label([WH time], i), label([product], i)=3), label([Expired], i):= 1, If(and(label([THT], i)<Label([WH time], i), label([product], i)=7), label([Expired], i):= 1)))`
Travel time (queue)	Capacity	`1000000`
Condition control (condition control)	Condition expression	`content(in(1, c))>0`
	Flow control on True	`Close input, allow output`
Storage (queue)	Capacity	`1000000`
	Send to	`if(Label([Expired], first(c)) = 1, 2, 1)`
	Trigger on entry	`Label([Distance],c):=Label([Distance],i)`
Expired products (queue)	Capacity	`1000000`
Order generator (source)	Inter-arrival time [s]	`86400*7`
	Time till first product [s]	`86400*6`
	Trigger on exit	`Label([Needed products], i):= dUniform(95, 110)`
Orders received (queue)	Capacity	`1000000`

Pick order (assembler)	Cycle time [s]	`0`
	Trigger on entry channel 1	`do(if(content(rank(54, Model))> 0, do(cell(2, 1, c):= label([needed products], i)-content(rank(54, Model)), cell(3, 1, c):= content(rank(54, Model))), do(Cell(2, 1, c):= label([needed products], i), Cell(3, 1, c):= 0)), Label([Exit time], i):= Time-Label([Entry time], i))`
Transport pallets (unpack)	Cycle time [s]	`0`
	Unpack quantity	`content(first(c))`
Recording data (data recorder)	Atom selection	`1. measure for all atoms`
	Analysis stop time [s]	`Hr(1000000)`
Assembly line (queue)	Capacity	`10000000`
	Send to	`if(Label([THT], first(c)) < 30*86400, 1, if(Label([THT],first(c))>70*86400, 3, 2))`
<30 (queue)	Capacity	`1000000`
	Trigger on entry	`Label([WH time],c):=Label([WH time],i)`
30-70 (queue	Capacity	`1000000`
	Trigger on entry	`Label([WH time],c):=Label([WH time],i)`
>70 (queue)	Capacity	`1000000`
	Trigger on entry	`Label([WH time],c):=Label([WH time],i)`

B. Model formulas that change per scenario

Table 5 Formulas in the model that differ per scenario.

Scenario	Atom name (characteristic)	Where in atom	Code
1, 2	Ready for storage place (queue)	Send to	4. A random open channel: choose a random channel from all the open output channels.
3, 4		Send to	Label([Destination], first(c))
5, 6		Send to	Label([Destination], first(c))
1, 2		Trigger on entry	-

3, 4		Trigger on entry	do(if(and(Label([THT], i)<= 30*86400, Label([THT], i)>= 0),
			(do(if(content(rank(Parameter(36, 1), model))< Parameter(24, 1),
			Label([Destination], i):= 6),
			if(content(rank(Parameter(36, 1), model))>= Parameter(24, 1),
			Label([Destination], i):= 5),
			if(and(content(rank(Parameter(36, 1), model))>= Parameter(24, 1),
			content(rank(Parameter(35, 1), model))>= Parameter(22, 1)),
			Label([Destination], i):= 3),
			if(and(content(rank(Parameter(36, 1), model))>= Parameter(24, 1),
			content(rank(Parameter(35, 1), model))>= Parameter(22, 1),
			content(rank(Parameter(33, 1), model))>= Parameter(22, 1)),
			Label([Destination], i):= 1)))),
			
			(if(and(Label([THT], i)<= 70*86400, Label([THT], i)> 30*86400),
			do(if(content(rank(Parameter(37, 1), model))< Parameter(24, 1),
			Label([Destination], i):= 7),
			if(content(rank(Parameter(37, 1), model))>= Parameter(24, 1),
			Label([Destination], i):= 9),
			if(and(content(rank(Parameter(37, 1), model))>= Parameter(24, 1),
			content(rank(Parameter(39, 1), model))>= Parameter(24, 1)),
			Label([Destination], i):= 4),
			if(and(content(rank(Parameter(37, 1), model))>= Parameter(24, 1),
			content(rank(Parameter(39, 1), model))>= Parameter(24, 1),

		content(rank(Parameter(34, 1), model))>= Parameter(24, 1)),
		Label([Destination], i):= 11)))),
		
		(if(and(Label([THT], i)<= 10000*86400, Label([THT], i)> 70*86400),
		do(if(content(rank(Parameter(43, 1), model))< Parameter(24, 1),
		Label([Destination], i):= 13),
		if(content(rank(Parameter(43, 1), model))>= Parameter(24, 1),
		Label([Destination], i):= 8),
		if(and(content(rank(Parameter(43, 1), model))>= Parameter(24, 1),
		content(rank(Parameter(38, 1), model))>= Parameter(26, 1)),
		Label([Destination], i):= 2),
		if(and(content(rank(Parameter(43, 1), model))>= Parameter(24, 1),
		content(rank(Parameter(38, 1), model))>= Parameter(26, 1),
		content(rank(Parameter(32, 1), model))>= Parameter(25, 1)),
		Label([Destination], i):= 10),
		if(and(content(rank(Parameter(43, 1), model))>= Parameter(24, 1),
		content(rank(Parameter(38, 1), model))>= Parameter(26, 1),
		content(rank(Parameter(32, 1), model))>= Parameter(25, 1),
		content(rank(Parameter(40, 1), model))>= Parameter(24, 1)),
		Label([Destination], i):= 12),
		if(and(content(rank(Parameter(43, 1), model))>= Parameter(24, 1),
		content(rank(Parameter(38, 1), model))>= Parameter(26, 1),
		content(rank(Parameter(32, 1), model))>= Parameter(25, 1),
		content(rank(Parameter(40, 1), model))>= Parameter(24, 1),
		content(rank(Parameter(42, 1), model))>= Parameter(24, 1)),
		Label([Destination], i):= 14))))
		)

5, 6	Trigger on entry	do(if(Label([product], i)=1, Label([Destination], i):= 14),
		if(Label([product], i)=2, Label([Destination], i):= 14),
		if(Label([product], i)=3, Label([Destination], i):= 12),
		if(Label([product], i)=4, Label([Destination], i):= 2),
		
		if(Label([product], i)=5,
		do(if(content(rank(Parameter(43, 1), model))< Parameter(24, 1),
		Label([Destination], i):= 13),

			if(content(rank(Parameter(43, 1), model))>= Parameter(24, 1),
			Label([Destination], i):= 8),
			if(and(content(rank(Parameter(43, 1), model))>= Parameter(24, 1),
			content(rank(Parameter(38, 1), model))>= Parameter(26, 1)),
			Label([Destination], i):= 10))),
			
			if(Label([product], i)=6,
			do(if(content(rank(Parameter(39, 1), model))< Parameter(24, 1),
			Label([Destination], i):= 9),
			if(content(rank(Parameter(39, 1), model))>= Parameter(24, 1),
			Label([Destination], i):= 4),
			if(and(content(rank(Parameter(39, 1), model))>= Parameter(24, 1),
			content(rank(Parameter(34, 1), model))>= Parameter(24, 1)),
			Label([Destination], i):= 11))),
			
			if(Label([product], i)=7,
			do(if(content(rank(Parameter(36, 1), model))< Parameter(24, 1),
			Label([Destination], i):= 6),
			if(content(rank(Parameter(36, 1), model))>= Parameter(24, 1),
			Label([Destination], i):= 5),
			if(and(content(rank(Parameter(36, 1), model))>= Parameter(24, 1),
			content(rank(Parameter(35, 1), model))>= Parameter(22, 1)),
			Label([Destination], i):= 3),
			if(and(content(rank(Parameter(36, 1), model))>= Parameter(24, 1),
			content(rank(Parameter(35, 1), model))>= Parameter(22, 1),
			content(rank(Parameter(33, 1), model))>= Parameter(22, 1)),
			Label([Destination], i):= 1),
			if(and(content(rank(Parameter(36, 1), model))>= Parameter(24, 1),
			content(rank(Parameter(35, 1), model))>= Parameter(22, 1),
			content(rank(Parameter(33, 1), model))>= Parameter(22, 1),
			content(rank(Parameter(31, 1), model))>= Parameter(23, 1)),
			Label([Destination], i):= 7))))

1, 3, 5	A1 till G2 (queue)	Queue discipline	1. Fifo (First In First Out)

2, 4, 6	A1 till (queue)	G2	Queue discipline	`t-findqueuepos([THT], 2)`

C. Model parameters

Table 6 Model parameters.

Parameter number	Parameter		Explanation
1	0.31	m/s	A1
2	0.19	m/s	A2
3	0.35	m/s	B1
4	0.25	m/s	B2
5	0.38	m/s	C1
6	0.64	m/s	C2
7	0.29	m/s	D1
8	0.2	m/s	D2
9	0.26	m/s	E1
10	0.19	m/s	E2
11	0.25	m/s	F1
12	0.18	m/s	F2
13	0.22	m/s	G1
14	0.17	m/s	G2
15	0.001702	%	Eggs
16	0.073959	%	Other
17	0.15318	%	Bread
18	0.245242	%	Coffee and tea
19	0.462788	%	Groceries
20	0.697509	%	Grain and rice
21	1	%	Potatoes, vegetables, and fruit
22	26		Pallet capacity
23	38		Pallet capacity
24	40		Pallet capacity
25	48		Pallet capacity
26	54		Pallet capacity
27	300	s	Cycle time forklift driver
28	300	s	Cycle time administration
29	420	s	Cycle time forklift driver
30	420	s	Cycle time administration
31	21		A1
32	22		A2
33	23		B1
34	24		B2
35	25		C1
36	26		C2
37	27		D1
38	28		D2
39	29		E1
40	30		E2

41	31	F1
42	32	F2
43	33	G1
44	34	G2

D. Capacity and distance of storage locations

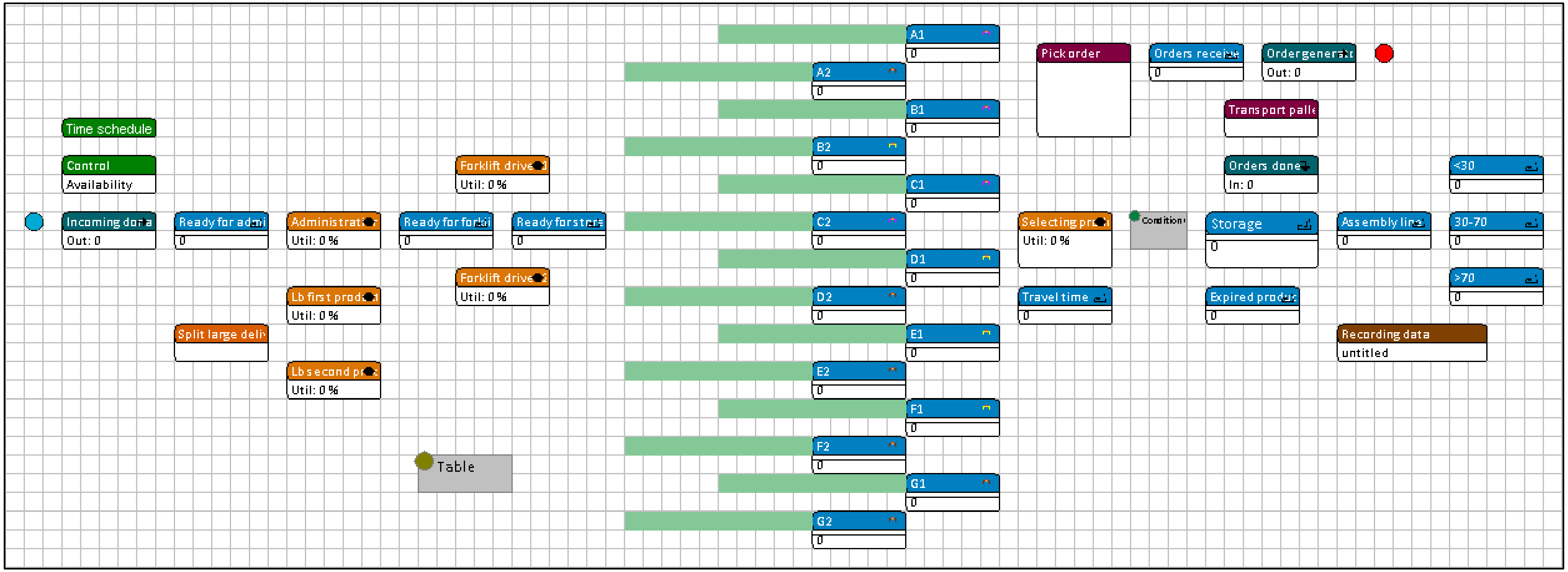
Table 7 Capacities per storage location.

Storage location	Capacity
A1	40
A2	50
B1	28
B2	42
C1	28
C2	42
D1	42
D2	56
E1	42
E2	42
F1	42
F2	42
G1	42
G2	56

Table 8 Speed of the fast non-accumulating conveyor based on the distance of the storage location.

Storage location	Speed
A1	0.31
A2	0.19
B1	0.35
B2	0.25
C1	0.38
C2	0.64
D1	0.29
D2	0.2
E1	0.26
E2	0.19
F1	0.25
F2	0.18
G1	0.22
G2	0.17

E. Simulation model food bank without channels



F. Simulation model food bank with channels