Solving the on-line Order Batching Problem
a case study

Janssen, J.

Award date:
2021

Solving the on-line Order Batching Problem: a case study

Eindhoven University of Technology

Industrial Engineering and Innovation Sciences

Jet Janssen (1373439)

Supervisors:

Dr. T. Tan
Dr. Z. Atan

Company Supervisor:
X

3rd Assessor:
Dr. Ir. R.A.C.M. Broekmeulen

Tilburg, May 2021

Keywords: order picking, Order Batching Problem (OBP), On-line OBP, priority rules, seed heuristic, warehousing, zoning

It should be noted that all numbers and costs mentioned in this thesis are fictitious due to confidentiality reasons

Abstract

As there is an increasing pressure for warehouses to deliver more orders in a shorter amount of time, it is vital for companies to improve their pick performance. This research investigates if the pick performance can be improved in a low-level, manual, picker-to-parts warehouse by switching from a single order picking policy to a batching policy. This is researched by a case study conducted at Company X. Four batching algorithms were designed to solve the Order Batching Problem (OBP). One algorithm was based on priority rules and the other three on seed heuristics. Orders that arrive at the warehouse must be picked on the same day. It is unknown in advance how many orders there will be and what kind of products are ordered. Therefore, this problem can be classified as an on-line order batching problem. To solve the on-line problem, the day was divided into several timeframes and within each timeframe the classical off-line problem could be solved. The algorithms were tested for different implementation options: for picking all the boxes or only boxes with a single pick line and for two different routing policies. The investment costs of the different options were also considered in this research. The results showed that batching can improve the pick performance irrespective of the chosen algorithm. The first algorithm, based on priority rules, leads to the highest saving in pick time.

Preface

This master thesis marks, not only the end of my master Operations Management & Logistics at the TU in Eindhoven, but also the end of my student life. Now I can look back on an incredible seven-year experience where I studied at Tilburg University, Jönköping Business School, Instituto Tecnológico de Buenos Aires, and Eindhoven University of Technology.

I would like to take this opportunity to thank several people who guided me during my masters. First of all, I would like to thank my mentor, Tarkan Tan, for the guidance and feedback during the master thesis project. Second, I would like to thank Zümbül Atan for being my mentor in the first phase of my master and for providing me with feedback as the second supervisor of my master thesis. Moreover, I would like to thank my company supervisor and colleagues for their support and guidance.

Furthermore, I would like to thank my parents for always supporting me and letting me make my own decisions. And finally, my friends who supported me throughout my student life.

Jet Janssen

Executive summary

This research is conducted at Company X. This is part of Company X-Y, which is one of the largest wholesaler in the world for IT and mobility products. The warehouse of company X is located in the south of the Netherlands and it is the distribution centre for Benelux.

Problem definition

There is an increasing pressure for warehouses to deliver more orders in a shorter amount of time. This also affects Company X. Customers do not only demand shorter lead times, but they also want more variety and place smaller orders. Therefore, it is important that their picking system can competitively handle all the incoming orders. The most time-consuming step in the picking process is travel time (van Gils et al., 2018). Therefore, this is one of the most important aspects when reducing the overall picking time. This leads to the following research question:

"*How can the pick performance in a low-level, manual, picker-to-parts warehouse be improved?"*

This research focusses on implementing order batching in a warehouse, where smaller products are located that have to be picked in master cartons. The physical layout of this area is out of scope as well as the storage location assignment, which is currently done by the ABC storage policy. The warehouse can be described as a low-level, manual, picker-to-parts warehouse. The warehouse is sub divided into zones and all the zones are connected by a conveyor system. The conveyor creates the boxes in which orders have to be picked and transports them to the zone. Boxes are picked with a single order picking process. This means that currently, there is no batching in the warehouse.

Moreover, as there is no significant investment budget available for new capital assets and new staff, this research focusses on the functionalities available at the new Warehouse Management System (WMS), and how these can be improved in a low-cost manner.

Modelling

In this research the Order Batching Problem (OBP) is solved by minimizing the pick time per batch. Four algorithms were developed based on heuristics commonly used to solve the OBP. The first algorithm, Algorithm 1 uses a combination of priority rules, namely the Earliest Due Date (EDD) and the First-In-First-Out (FIFO) rule. The other three algorithms are based on seed heuristics. Algorithm 2 makes batches based on the nearest aisles, Algorithm 3 on the closest distance and in Algorithm 4 only boxes with pick lines in the same aisle are allowed to be batched. To be able to measure the performance of the algorithms, they are applied as a case study to Company X using production data of 12 days in February and March 2021.

As all orders that arrive during the day have to be picked the same day, the OBP is an on-line problem. The problem is solved by dividing the day into several timeframes. Orders that arrive within a certain timeframe will be saved and released after the timeframe has ended.

To be able to implement the algorithms, two implementation options for Company X need to be considered. The first option is if all the boxes can be batched or only the single line boxes.

With the new WMS boxes can only be added to one specific pick cart but one pick cart cannot travel between zones. Therefore, boxes with picks in several different zones cannot be batched as the carts cannot move across different zones. A change request can be made to add a function to remove boxes from the pick cart. However, this will take some additional investment. The second aspect concerns the routing. The routing currently in place is very inefficient. A new routing policy is proposed, that resembles the S-shape heuristic. Implementing this new routing policy will also lead to more investment costs. These two implementation options are taken into account by developing four options for how the batch picking can be implemented. Option 1 only batches the single line boxes for the routing currently in place in the warehouse. This is the most basic option and can be implemented right away and without any additional investments or changes in the warehouse. Option 2 also batches only single lines, but it creates the routes according to the new routing policy. Option 3 batches all the boxes but for the current routing and lastly, Option 4 combines batching all the boxes and the new routing policy.

Results

The algorithms in the case study are simulated for batch capacities of 9 and 12 boxes, as these are the most feasible capacities for the warehouse layout of the company. When the batch capacity increases the percentage of savings increases as well, as a higher capacity increases the pick density because more pick lines can be combined in one route. For a capacity of 9 boxes Algorithm 4 is the best performing one under all implementation options. However, the performance of all four algorithms is very close. For a capacity of 12 boxes Algorithm 1 clearly performs best. The proposed routing policy performs slightly better than the current routing policy, as pick locations will be further apart due to the random formation of batches for Algorithm 1, this algorithm will benefit more from a logical routing. Moreover, batching all the pick lines performs better than only batching the single line boxes, as more boxes can be batched and therefore more and better combinations can be formed. Combining the new routing policy with batching all the pick lines gives the best overall performance. The percentages increase in performance for the best performing algorithms per capacity can be seen in Table 1.

Table 1: Savings per implementation option

	Option 1	Option 2	Option 3	Option 4
A4 – 9 boxes	4,26%	4,55%	5,31%	6,04%
A1 – 12 boxes	5,26%	5,69%	7,00%	7,77%

Several sensitivity analyses were performed and showed that the algorithms are robust for changes in the input parameters of the model. Only for batch capacities smaller than 9 boxes the results of the algorithms change. Algorithm 1 performs poorly for small batch capacities. Also, when there are no timeframes the other algorithms outperform Algorithm 1 when there is a higher batch capacity. However, these two situations are very unlikely to happen as higher batch capacities always outperform smaller capacities, so the company will go for a pick cart with a capacity of at least 9 boxes, and the company will not be able to switch to an off-line problem due to the nature of the business to deliver orders as fast as possible.

In general, the performance of the algorithms slightly increases when the number of pick lines increase, an increase in travel time leads to an increase in savings and an increase in set-up time to a decrease in savings. In a scenario analysis the robustness of the algorithms on the warehouse layout is tested and concluded that they are robust for changes in the characteristics of the warehouse layout.

The costs were calculated per implementation option and batch capacity. From these costs the payback time and Net Present Value (NPV) (for a 10-year period, without considering the interest rate) were determined. From this calculation it was concluded that for a capacity of 9 boxes the savings of the batching all boxes function do not outweigh the extra investments. The savings of the new routing policy did outweigh the investments. Therefore, Option 2 was considered the best option for this capacity both in terms of payback period and NPV. For a capacity of 12 boxes, Option 2 had the lowest payback period. However, taking into account the NPV, Option 4 became the best. If Option 4 is implemented with a capacity of 12 boxes the required investments would be 59.865 euro and the savings are almost 10.500 euro per year. The required investments for this option consist of purchasing pick carts, implementing the new WMS, implementing several change requests for this WMS and changing the routing policy.

Recommendations

The recommended batch capacity is 12 boxes, as this is a feasible capacity for the warehouse and leads to a higher savings in pick time than a capacity of 9 boxes. For a capacity of 12 boxes the algorithm that leads to the highest savings is Algorithm 1. As expected, the option (Option 4) in where all the pick lines are batched, and the new routing policy is implemented performs best. Even considering all the additional investments needed it is still the preferred option in terms of NPV. However, the changes required to implement Option 4 will take some time and effort to implement. Therefore, it is recommended to start with Option 1. This means that only the single line boxes are batched, and the routing policy remains unchanged. Once Option 1 is in place and working properly, the company can gradually implement a change in their routing policy. It is recommended to do this zone by zone, so potential mistakes in one zone can be identified are not copied to all the zones. A change request can be made at the software department to add the function to batch all pick lines. This will take some time to complete. The company could also search for other sites interested in this change, so the investment costs can be split over multiple sites. This would decrease the investment costs for the company and decrease the payback period and increase the NPV of Option 4 significantly.

Changing the sequence in which the orders are picked is connected to the logic in which products are stocked by the outbound department. Therefore, it is recommended to thoroughly investigate the implications of changing the routing policy on this stocking logic, so this process is not negatively affected by changing the routing policy.

For future research it is recommended to investigate the effects of integrating zoning with other batching methods and to focus more on the on-line batching problem.

Contents

List of figures

List of tables

Glossary

3PL	Third party logistics
ABC storage assignment	Form of class-based storage in which the products are divided into three categories: A, B and C.
Batch	A group of orders or pick lines to be picked in one single pick tour.
Bay	Storage rack. A bay is subdivided into several bins.
Bin	A single storage location in the warehouse. A bin can store several units of the same product.
Box	Piece of carton in where the products are picked and shipped to the customer.
Class-based storage	Divides all products in the warehouse in classes based on demand rates. Products with higher turnovers are located closer to the I/O point.
Cluster pick	See batch
Conveyor	Belt that transports boxes throughout the warehouse.
Cross-aisle	Aisle perpendicular to the storage aisles.
FIFO	First In First Out
I/O point	Point where the batch tour starts and ends; point where an empty box is retrieved, and a full box is returned.
KPI	Key Performance Indicator
Multi-line box	A box that has multiple pick lines.
Order congruency rule	Rule that determines which box is added to the batch.
Order line	Division of one order into different SKUs. One order line can be divided into several pick lines.
Pick	Retrieving a product from the storage location.
Pick cart	Device on which multiple boxes can be placed which is used for a pick tour.

Pick line	Quantity of SKUs that have to be picked from a single location.
Pick tour	The route to be travelled to complete all the picks in the batch.
Picker-to-parts	Warehouse configuration in where the people walk to the products.
Prep desk	Department that is responsible for assigning the orders to the right pick department.
Repack area	Storage area where smaller products are stored.
RF scanner	Radio Frequency scanner. Is used to scan products in the warehouse.
Routing	Formation of routes.
SAP	Enterprise resource planning (ERP) software used in the warehouse.
Seed order	First order that is added to the batch.
Sequential zoning	Zoning system in where all products required for an order are picked sequentially.
Single-line box	A box that only has one pick line.
Single order picking policy	Picking policy in where each box is picked separately.
SKU	Stock Keeping Unit. A distinct type of item for sale.
Timeframe	A period of time.
Two-sided picking	Warehouse configuration in where both sides of the aisle can be picked simultaneously without additional walking time.
WMS	Warehouse Management System
Zone	A subdivision of the warehouse area. Consists of several aisles.

1. Introduction and problem formulation

In this chapter the company will be introduced shortly in Section 1.1. In Section 1.2 the project context is given. Section 1.3 defines the problem and states the research goal. The research questions are posed in Section 1.4 and in Section 1.5 the scope of the research is defined. The methodology to solve the research questions is explained in Section 1.6. Finally, in Section 1.7, an overview is given of the structure of the remainder of the report.

1.1 Company introduction

Company X-Y is one of the largest wholesaler in the world for IT and mobility products. Its headquarters are located the United States. It has logistic centres all over the word, local sales offices in numerous countries and offers its services in many countries around the world. The company is divided into several business units. The Dutch head office is located in the centre of the Netherlands. Furthermore, there are several warehouses in the south of the Netherlands.

This master thesis will focus on one of these warehouses in the south of the Netherlands, Company X (also referred to as 'the company' or 'the warehouse' in this report). This warehouse is the distribution centre of the Benelux. Daily, they receive IT and mobility products from over 80 vendors and ship it to more than 3460 customers in mainly the Netherlands and Belgium, but also to some other European countries.

1.2 Project context

Warehouses are under an increasing pressure to deliver orders on time and preferably on the same day as they received the order. Moreover, the product variety in the warehouse increases and customers order more frequently and in smaller quantities. Therefore, it is important that picking system in warehouses are able to competitively handle all the incoming orders. To fulfil these orders, the stock keeping units (SKUs) have to be picked by operators in a warehouse. The picking process is very time and labour intensive. According to de Koster et al. (2007), picking costs can account for as much as 55% of the total warehouse costs. Therefore, it is important for warehouses to reduce the picking time as much as possible. Less time spent on picking will lead to less labour costs and more orders that can be delivered on time.

The picking process can be divided into several components: setup, travel, search, pick, sort and idle time. Of these components, travel time is the most time consuming (van Gils et al., 2018). Reducing the travel time is therefore one of the most important aspects to reduce the overall pick time.

According to van Gils et al. (2018) order picking decisions can be made on three levels: strategic, tactical and operational level. Strategic decisions are long term decisions. They focus on the layout of the warehouse, the selection of the storage system and the material to collect the SKUs from their storage locations. Tactical decisions impact the medium term. Examples are dividing the warehouse into zones and assigning the SKUs to a location in the warehouse. The last level, operational, focusses on the daily decisions that need to be made, like batching and routing. This research will focus on the operational decision batching and how this decision can improve the pick performance.

1.3 Problem definition and research goal

At Company X, incoming orders can be processed in three ways: as pallet picks, full case picks or in the repack area. If the order is sufficiently large, the order is picked as a pallet. Orders that require SKUs that are large enough to go over the conveyor without a master carton are picked as full cases. This entails that they are not picked at the repack area, but at another location in the warehouse. They can be placed on the conveyor belt but are not transported through the entire repack area. They only go to the billing and truck scan stage of the conveyor.

In the repack area of the warehouse smaller SKUs are located that need to be shipped with a master carton. This area can be described as a low-level manual picker-to-parts warehouse. The area consists of two floors and is divided into 33 zones. The zones are connected by a conveyor system. Each zone has an Input/Output point (I/O point; point where an operator can take an empty box and return a full box) on the conveyor belt. From the beginning to the end of the conveyor belt, the SKUs become smaller and lighter. This is done to make sure the heavy products are packed first and are at the bottom of the shipping box. The zoning system is sequential, meaning a box is passed from zone to zone to complete a customer's order (van Gils et al., 2018). The boxes are picked with a single order picking policy, meaning that one box at a zone must be finished before the picker starts on the next box. In the repack area, the more experienced operators already put aside boxes with locations far from the I/O point. When they have collected several boxes, they pick them all together to save walking time. This indicates that the process could be improved by batching at least part of the orders. However, this has never been investigated and there is no formal system in place for this kind of batching.

The current Warehouse Management System (WMS) does not provide any option for batching orders. If the company changes their WMS to a new version, orders can be batched. With this new WMS, operators can place boxes on a cart and pick them as a batch. However, changing the WMS will lead to additional costs.

The goal of this research is to provide insights on how this batch option in the new WMS can influence the pick performance and what kind of investments are needed to implement this.

1.4 Research questions

Based on the problem definition and research goals the following research question (RQ) can be defined:

"*How can the pick performance in a low-level, manual, picker-to-parts warehouse be improved?*"

To be able to answer this research question, the following sub questions (SQ) are stated:

1. *How can the current picking process in the repack area be described?*
2. *What is the current performance of the picking process in the repack area?*
3. *What are suitable batch and cluster methods to improve the pick performance in the repack area?*
4. *How can the performance of these methods be calculated?*
5. *What is the expected improvement from the new methods?*
6. *What are the expected required investments for the new methods?*

1.5 Scope

The scope of the thesis project focuses on the repack area of the warehouse, so the other outbound divisions and other departments like receiving and returns are out of scope.

The report focusses on the operational level order picking decision: batching. The strategic decision, the physical layout of the repack area, is out of scope. Hence, the location and quantity of the storage racks is fixed. Also, the tactical decisions are out of scope. The storage location assignment is given. SKUs are currently stocked according to the ABC logic with a diagonal location policy (van Gils et al., 2019). This policy is recently investigated and implemented. Therefore, there is no need to reinvestigate this in this thesis project.

Moreover, there is no significant investment budget available for new capital assets and new staff. Therefore, the research focused on the functions available at the new WMS, and how these can be improved in a low-cost manner.

The actual implementation of the suggested new methods is out of scope. This is due to the limited time frame of the master thesis project.

1.6 Methodology

The company uses the DMAIC structure for their projects. This structure consists of five phases: Define, Measure, Analyse, Improve and Control. In the Define phase the project and the scope of the project are defined. In the Measure phase the data necessary to execute the research is collected. In the Analyse phase the data collected in the Measure phase is analysed and root causes are identified. During the Improve phase solutions are suggested and validated. Finally, the Control phase focuses on how the improvements can be sustained[1]. Every sub question posed in Section 1.4, is answered in one of the DMAIC phases.

Define:

What is the approach of the master thesis project?

The approach for the project is described in this chapter, Chapter 1. This question is not included as a separate sub question. In this phase other employees of the company were

[1] DMAIC - The 5 Phases of Lean Six Sigma. Available online: https://goleansixsigma.com/dmaic-five-basic-phases-of-lean-six-sigma/ retrieved at 19-11-2020

informed about the project. This was done to create awareness for the project and to let employees know when what kind of information was needed from them.

Measure:

1. *How can the current picking process in the repack area be described?*

The first research question is answered during the Measure phase of the project. In this phase all the steps of the picking process in the repack area were visualised by making a business process diagram. This question is included to get a clear view and good understanding of the current picking process. Moreover, in this phase the general process and warehouse layout of the warehouse is described and the composition of the orders in the repack area was analysed.

2. *What is the current performance of the picking process in the repack area?*

This question is answered to establish a baseline of the current performance at the warehouse. This baseline is needed to be able to determine if the new suggested methods can improve the performance. To be able to answer this research question the current process is analysed. The analysis of the current process was done in two ways. The first way is to analyse the process using data retrieved from the data management system (DMS) of the company. The second way is an empirical study. Value Stream Mapping (VSM) is used to gain insights in the picking process and the time every step takes in this process. The reason for combining an empirical study with data from the database is that the exact travel time cannot be retrieved from the database. With the empirical study an estimation can be made on the time spent on travel per pick. The performance does not only include the time spend on travel but also the quality of the process.

Analyse:

In the Analyse phase, first SQ3 is answered:

3. *What are suitable batch and cluster methods to improve performance in the repack area?*

Common batching methods in scientific literature were identified and analysed to see if they are applicable to the repack area. The proposed methods have to be practically implementable in the WMS without too much effort and cost. Moreover, the methods were analysed to see if they are applicable for all types of SKUs or only for slow- or fast-moving SKUs and if they can be best applied to boxes with one or more pick lines. After the most suitable methods and SKUs are identified, the final question of the Analyse phase can be answered:

4. *How can the performance of these methods be calculated?*

A suitable method has to be found to calculate the performance of the proposed methods. This can be done for example by simulation. Also, different scenarios have to be identified, to see if the methods work in various settings. For example, days with fewer and days with more orders than usual. The answer for this question also includes the calculation of every method posed in SQ3.

Improve:

In the Improve phase the following questions are answered:

5. *What is the expected improvement from the new methods?*
6. *What are the expected required investments for the new methods?*

The first question, SQ5, is answered by comparing the performance of the simulation of SQ4 with the baseline performance determined in SQ2. To answer SQ6, the additional material required for the new methods is analysed and an estimation of the associated costs is given. Furthermore, the non-monetary investments are described, like the training of staff to be able to perform the new tasks or work with the new methods.

Control

This phase is not included in the scope of the master thesis. In this phase the company will need to decide on a method to implement in their warehouse. The actual improvement of the chosen method must be measured and compared to the simulated improvement of the Improve phase. Furthermore, in this phase it is reviewed how the improvement can be sustained.

1.7 Report structure

The remainder of the report follows the DMAIC structure. The measure phase consists of Chapter 2 in which the case study environment and the current processes are described. Chapter 3 and 4 make up the Analyse phase. In Chapter 3 an overview of the relevant literature regarding order batching is discussed and suitable approaches for this research are identified. Chapter 4 gives the model and all the input variables, assumptions and decisions related to the model. The Improve phase consists of Chapter 5, 6, 7 and 8. In Chapter 5 the results of the model are discussed. In Chapter 6 and 7 a sensitivity and scenario analysis are conducted to see if the model is robust to changes in the parameters or warehouse layout. Chapter 8 gives an overview of the investments needed to implement batching in the warehouse. Lastly, a conclusion is given in Chapter 9, followed by recommendations for the company, the limitations of this research and interesting areas for future research.

2. Measure: Company characteristics

Chapter 2 is the Measure phase in the DMAIC process. The main goal of this phase is to collect all the data necessary to execute the research. In this chapter an overview is given of the features and characteristics of the company. First, in Section 2.1, the general process is described. In Section 2.2 the picking process is discussed in more detail. Section 2.3 describes the order picking decisions that are in place at the company and finally, in Section 2.4, the current performance is measured and described. Section 2.5 gives a short conclusion about the findings in this chapter.

2.1 General process

Trucks with products from vendors are received and unloaded at the inbound department of the warehouse. Inbound operators check the shipments to see if the quantity and quality of the ordered SKUs is correct. Then the SKUs are booked in the system. Locations are assigned to the SKU and inbound operators stock the SKUs in the right location.

Incoming customer orders are processed in SAP. SAP connects with the WMS of the warehouse and sends the orders through. In the WMS the orders can be prepared and released to the warehouse. This is the responsibility of the prep desk. The prep desk decides how to release the orders. This can be done in three ways: pallet pick, full case or repack. The way an order is released depends on the quantity and characteristics of the requested SKUs. Pallet pick is used when an order consists of more than 20 units of the same full case SKU or the weight of a single SKU exceeds 32 kg. SKUs that are too big to be transported on the conveyor in a master carton are picked as full cases. The last option is the repack area. In this area, smaller SKUs are located that need to be shipped with a master carton. Different SKUs can be packed in the same box. After the orders are released by the prep desk, the assigned inbound team can pick them.

The pallet and full case team processes the orders that are released for pallet pick or full case. The SKUs are picked at the warehouse and manually labelled and confirmed. Orders that are released as repack go over the conveyor. For these orders, the operators pick the products in the repack area. Boxes are created, labelled and confirmed automatically. Complete orders are loaded into trucks of third-party logistics providers (3PL) and transported to the customer. All customer orders that arrive before the cut off time of 18:30h must be processed on the same day.

When customers return products, they are processed by the return department. If the products are unopened and undamaged, they can be sold again and are booked on stock and restored in the warehouse. Products that are damaged or opened are placed in a separate location in the warehouse. These products are returned to the vendor or sold with discount. If the products cannot be sold with a discount they are written off and thrown away. A visual representation of the general process at the warehouse can be seen in Figure 1.

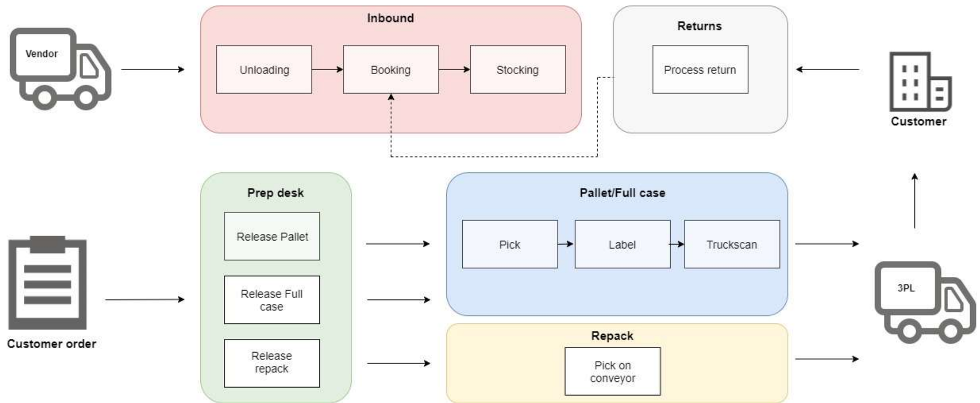


Figure 1: General process at the warehouse

2.2 Picking process

This section describes the picking process of the repack area in more detail. The picking process starts with releasing the orders (Section 2.2.1). After they are released, the orders can be picked (Section 2.2.2).

2.2.1 Order release method

Customer orders arrive during the day in SAP. SAP sends these orders through to the WMS. At the beginning of the day, there can still be some unreleased orders from the previous day in the system. These are usually orders that were requested by the customer after the cut off time of 18:30h. The other orders arrive during the day and the quantity and characteristics of the orders is unknown in advance.

The prep desk is responsible for releasing these orders to the warehouse. As mentioned in Section 2.1, they can release the orders in three different ways: pallet, full case or repack order. There are many variables a prepper needs to consider when deciding how to release an order. These variables include, but are not limited to, the 3PL that transports the order to the customer, the size and quantity of the SKUs, and if the customer already ordered more products that day. The system automatically calculates the number of boxes needed to fulfil the order and determines which SKUs are placed in which boxes. It also automatically divides the SKUs of a box into pick lines. A pick line is the quantity of SKUs that have to be picked from a single location. For example, a box can require the following products: three keyboards and four USBs. The four USBs can be picked from the same location. At the moment, there is no storage location that has three keyboards in stock. Therefore, two keyboards can be picked from location 1 and the last keyboard has to be picked from location 2. This box will have three pick lines, the first pick line is the four USBs from the same location, the second the two keyboards from location 1 and third, the last keyboard from location 2. Orders that are released as repack are sent through to the system that operates the conveyor.

2.2.2 Retrieval of orders

After orders are released by the prep desk, the conveyor creates the box or boxes for the orders. The boxes are transported over the conveyor belt and placed in the I/O point of the zone where the first SKU is requested. The picking process can be described as a single order picking policy (de Koster et al., 2007), as operators work on one order at the time. An operator takes a box at the I/O point and scans this box with his or her scanner. The scanner displays the location and the quantity of the required SKU. The operator leaves the box on the I/O point and walks to the location. At the location the SKUs are retrieved and scanned to confirm that they are picked. If required, the serial numbers are scanned as well. If there are more SKUs needed for the order, the scanner displays the next location after the previous order is picked. The operator checks if the location of this SKU is located in the same zone as he or she is operating and if this is the case, walks to the location and picks it as well. If not, he/she walks back to the I/O point and places the SKUs in the box. The operator is now finished with this box and places it back on the conveyor belt. The box is transported further over the conveyor and stops, if necessary, at other zones. When it has passed all the zones, it arrives at the incorrect-weight-lane station. Here, the actual weight of the box is checked with the weight according to the system. If these weights do not correspond, the box is reverted to a station where an operator checks the content of the box. If the content is correct, the box is placed back on the conveyor. If not, it is picked again by an operator. After this step, boxes are confirmed, closed, and labelled by a machine. Then they are transported to the docks, where they are loaded in the truck of the right 3PL. The process of the entire repack area is visualized in a business process diagram which can be seen in Appendix A. Appendix B contains a more detailed business process diagram of the pick process.

Boxes have, besides a barcode, a number between 1 and 8 printed on them. This number corresponds to a 3PL provider and indicates the priority of the box. There are two priority groups. The first group has the number 1, 2, 3 or 4 printed on them and have the highest priority. Therefore, they have to be picked first. Boxes of the second group have the number 5, 6, 7, or 8 and can be picked when there are no more boxes of the first priority group on the I/O point. The priority of boxes is determined by the time the 3PL provider picks up the boxes from the warehouse.

In the repack area, the operators start working at 13:00h. Operators are equally divided over the different zones of the repack area. Shift leaders keep track of the amount of work in each zone and redistribute operators accordingly.

2.3 Order picking decisions

According to van van Gils et al. (2018) order picking decisions can be made on three levels: strategic, tactical and operational level. Section 2.3.1 will discuss the strategic level decisions in the warehouse. Section 2.3.2 the tactical and finally, Section 2.3.3 the operational level decisions.

2.3.1 Strategic decisions

Strategic decisions are long term decisions. They focus on the selection of the storage system, the material to collect the SKUs from their storage locations and the layout of the warehouse.

Order picking system

According to de Koster (2012) order picking systems can be divided into two main groups: systems that employ humans and systems that employ machines. The machine group refers to systems with automated picking and picking robots. The human group is subdivided into picker-to-parts, put system and parts-to-picker. In a picker-to-parts warehouse, operators move to the SKUs that need to be picked. This is the case at the repack area. Furthermore, the top shelves of the storage bays can be reached without any additional equipment like a reach truck. Therefore, the repack area of the warehouse can be described as a low-level, manual, picker-to-parts warehouse. There is a sequential zoning system and as operators work on one order at the time, there is a single order picking policy.

Layout warehouse

The repack area covers an area of 9.000 m2 and consists of two floors. SKUs are stored in bins. SKUs that are stored in this area can have a maximum volume of 96 litre and a weight of 20 kilograms. There are over 20.000 bins storing over 14.200 types of SKUs. Around 80% of the bins are occupied. In total, there are more than 300.000 products stored in the repack area. Every bin has its own unique number and barcode.

The storage racks are placed close together and therefore the warehouse layout supports two-sided picking. This means that there is no additional travel time when switching from picking products on the right side or on the left side of the aisle (Scholz & Wäscher, 2017). However, currently the numbering of the locations does not support this two-sided picking. The numbering can be changed so two-sided picking can be implemented, although this will require a substantial amount of time. The aisles are divided into zones, which will be explained in the next section, about tactical decisions.

2.3.2 Tactical decisions

Tactical decisions impact the medium term. Examples are dividing the warehouse into zones and assigning the SKUs to a location in the warehouse.

Zones

The repack area is divided into 33 zones. As already mentioned, the zones are connected by a conveyor system and each zone has an I/O point on the conveyor belt. From the beginning to the end of the conveyor belt, the SKUs become smaller and lighter. This is done to make sure the heavy products are packed first and are at the bottom of the shipping box. The zoning system is sequential, so a box is passed from zone to zone to complete a customer order and no consolidation is needed at the end of the process (van Gils et al., 2018). The downstairs area has 18 and the upstairs area 15 zones. The layout of a single zone can be described as a rectangular, single block warehouse with a cross aisle at the front and back of the zone. The zones have storage bays. On these bays storage locations or bins are created. Bays have two, four or seven shelves. Bays with more shelves are mainly located in the zones at the end of the conveyor as they are more suited for smaller products. An aisle usually has a depth of three or four bays. A zone consists on average of seven aisles. A map of the upstairs floor of the repack area can be seen in Figure 2. Each colour on the map of Figure 2 represents one of the 15 zones in the upstairs area.

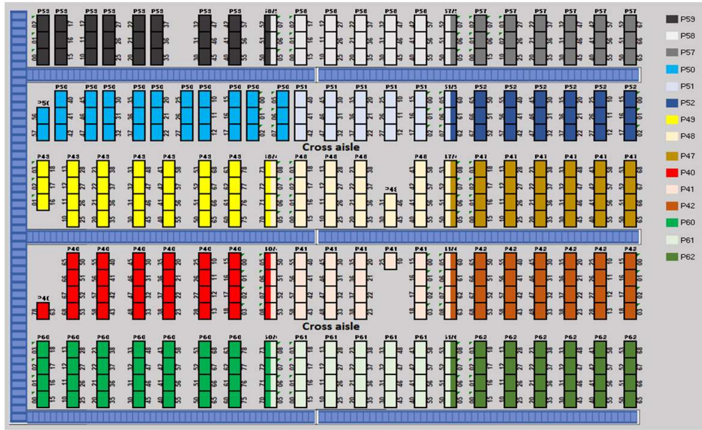


Figure 2: Map of the repack area (upstairs)

Storage location assignment

Storage location assignment is allocating SKUs to storage locations in the warehouse (Kübler et al., 2020). This can be done randomly or by grouping SKUs into classes based on a specific characteristic, which is called class-based storage. The warehouse uses class-based storage. Items are divided into three classes (A, B and C), based on pick frequency. Class A contains SKUs with a pick frequency of 70% in the past 28 days. Class B a frequency of 30% and class C contains the products that have not been picked at all in the past 28 days. This results in generally 3% of the SKUs belonging to class A, 54% to class B and 43% to class C. SKUs belonging to the A class are placed on locations nearest to the I/O point. Then SKUs of class B and SKUs of class C are furthest from the I/O point. This is according to the diagonal storage location assignment policy (van Gils et al., 2019) and a single aisle contains therefore multiple classifications of SKUs. Every week the division of the SKUs is recalculated on Saturday. Products that enter the warehouse are stocked according to the new ABC classification, but products that are already in the warehouse are not relocated. New SKUs are always assigned to the A class.

It is possible that a SKU has several different locations in the repack area. This can be due to capacity limits of the bins or as a tactical decision to have some bins with full cases and other bins with single units of the same SKU.

2.3.3 Operational decisions

The last level, operational, focusses on the daily decisions that need to be made, like batching and routing. Currently, there is no order batching used in the repack area of the warehouse. Orders are picked by a single order picking policy. Therefore, there is also no picker routing policy in place. As operators only have to visit one location, they can just walk directly towards it. However, it can happen that a multi-line box requires multiple products from the same zone. If this is the case, the locations of the products will appear in an alphanumeric order on the Radio Frequency (RF) scanner of the picker.

2.4 Measure of performance

To be able to answer the second sub-question: *"What is the current performance of the picking process in the repack area?",* the key performance indicator (KPI) that is used for this research is discussed: the average daily pick line productivity per picking hour. This can be calculated by dividing the average number of daily picks of a given period by the average daily pick hours of that same period.

$$average\ daily\ pick\ line\ productivity\ = \frac{average\ daily\ \#\ pick\ lines}{average\ daily\ \#\ pick\ hours}$$

The average number of pick lines for 2020 was 5.992 per day in 67,7 picking hours per day This results in an average a pick line productivity of 88,5 pick lines per hour per operator for 2020. Even though 2020 was a strange year in many ways, for the company the workload was quite similar to that of the previous years.

In the next section, descriptive statistics will be given about the current amount of pick lines. In Section 2.4.1, general descriptive statistics will be given. The picking process will be decomposed in Section 2.4.2 to see how much time operators spend on every step in the process. Finally, the quality of the process is discussed in Section 2.4.3.

2.4.1 Descriptive statistics

In 2020 the mean number of pick lines in the repack area was 5.992. A histogram of all the daily pick lines in 2020 can be seen in Figure 3. From the histogram it can be seen that the distribution of the pick lines is not normally distributed, it is slightly skewed to the left. However, as it is only a slight deviation to the normal distribution, it is assumed for simplicity reasons that the daily pick lines do follow the normal distribution.

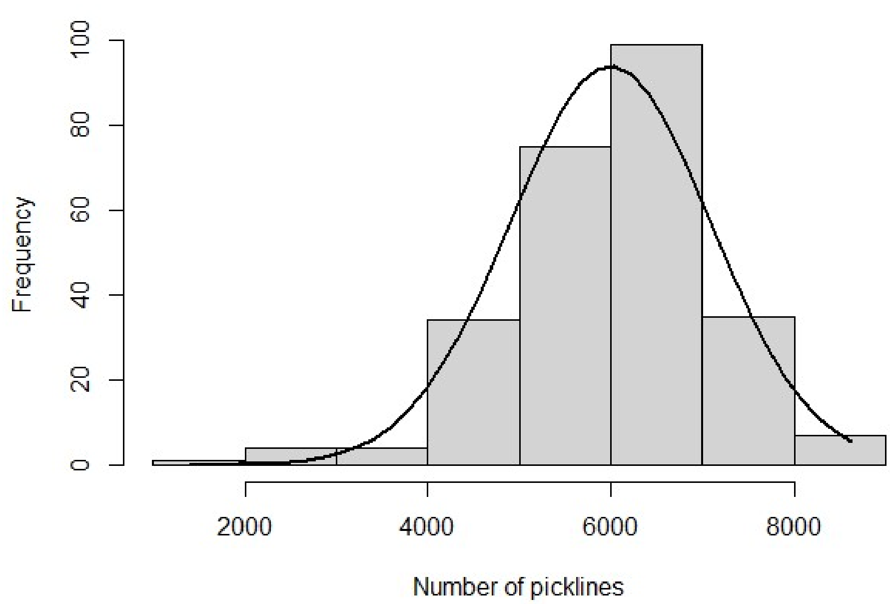


Figure 3: Histogram of the daily pick lines (2020) with normal curve

Other descriptive statistics concerning the amount of pick lines in 2020 can be seen in Table 2.

Table 2: Descriptive statistics pick lines 2020

Metric	Value
N	259
Mean	5.992
Median	6.102
Mode	6.257
Standard deviation	1.098,6
Maximum	8.633
Minimum	1.383

The company has over 3.400 customers. Not all these customers order on a daily basis. On 'Quiet' days the average number of customers is 614. On 'Busy' days 959 and on a 'Normal' day 893 customer orders are placed on average. A customer can place multiple orders on the same day. However, the majority of the customers, around 60%, only place one order. And most of these orders are relatively small. They only consist of one box and a one or a few pick lines. The other 40% of customers place multiple orders a day. However, these orders are divided into several boxes and each box is picked separately on the conveyor. Around 72% of all the boxes only have one pick line. This means that only one type of SKU from one location has to be picked. So even though the total customer orders might be large, the amount of products needed for a single box is usually small. According to de Koster et al. (2007), small orders indicate that there is a potential time saving when batching several orders in a single tour instead of using a single order picking policy.

Arrival of orders during the day

As mentioned in Section 2.2.1 orders are not known in advance and arrive during the day. Orders that arrive before 18:30h have to be processed the same day. Figure 4 shows the average number of boxes that arrive in the system per half hour. When the orders are available, the prep department can release them directly or leave them in the system to release at a later time. As can be seen in Figure 4, most orders arrive between business hours, between 8:30h and 18:30h. There are two peaks during the day. The first between 10:00h and 11:00h and the second at 16:00h. Each day follows a similar pattern regarding number of boxes per half hour.

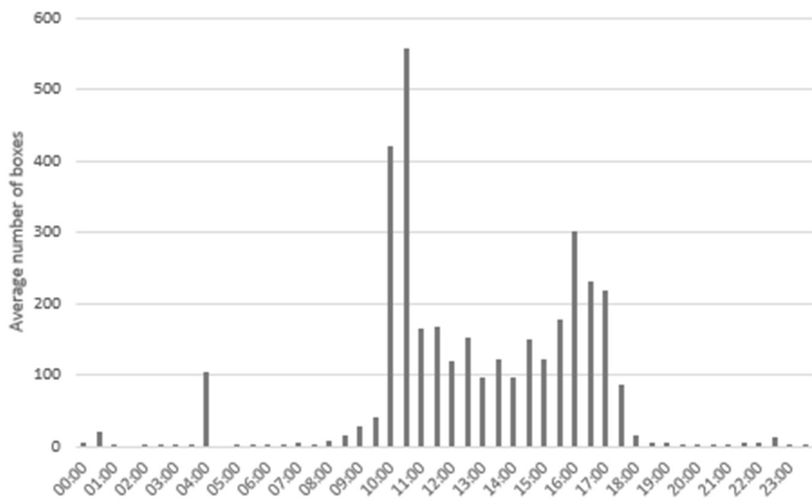


Figure 4: Average number of boxes per half hour

2.4.2 Time distribution picking process

The picking process consists of several different steps as explained in Section 2.2. In Appendix B a more detailed business process diagram is given of the steps relevant to this research. Every step of this process takes a certain amount of time. The process has been observed to get insight in the time distribution of every step. This was done by making videos of an operator during the pick process. In total 74 picks were analysed.

The result of these observations can be seen in Figure 5. An average pick in the repack area takes 90 seconds. Most of this time is spent on three steps: walking, taking and scanning the SKU. For the repack area the most time-consuming step is walking to the location of the SKU (28%). This is in line with the research of van Gils et al. (2018) who, as mentioned in Section 1.2, state that traveling time is the most time-consuming step in the picking process and therefore one of the most important aspects to reduce the overall picking time. It would also be interesting to see if the time needed to take and scan the SKU can be reduced, but this is out of the scope of this master thesis project. The category ‘other’ refers to other small tasks that are done during the picking process, like disposing empty boxes, wrapping hard drives in special boxes and getting tape.

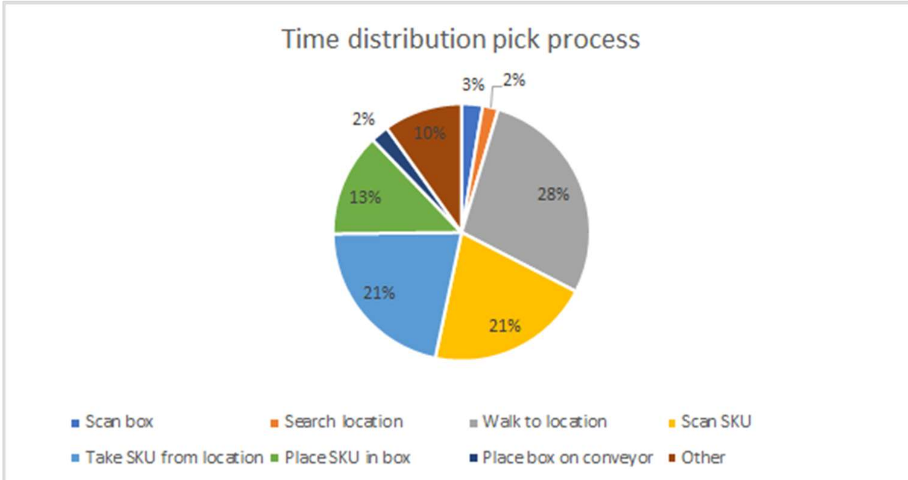


Figure 5: Time distribution of the picking process

To validate the time distribution, the 95% confidence intervals and the margins of error were calculated per step. These can be seen in Table 3. Especially the steps 'Scan SKU', 'Take SKU from location', 'Place SKU in box' and 'Other' have a wide 95% confidence interval, indicating that the estimate of the mean might not be very precise and reliable. This can be explained by the different quantities of SKUs that need to be picked for an order.

Table 3: Time distribution of the picking process (in seconds)

Step	Mean	SD	ME	Lower bound	Upper bound
				95% confidence interval	
Scan box	2,372	1,111	0,212	2,159	2,584
Search location	1,784	1,583	0,303	1,481	2,087
Walk to location	25,236	11,949	2,285	22,952	27,521
Scan SKU	18,649	25,210	4,820	13,828	23,469
Take SKU from location	19,399	28,113	5,376	14,023	24,774
Place SKU in box	11,716	21,741	4,157	7,559	15,873
Place box on conveyor	2,047	1,153	0,221	1,827	2,268
Other	8,919	19,548	3,738	5,181	12,657
Total	90,122	73,666	14,086	76,036	104,207

2.4.3 Quality of picking process

Quality is one of the most important performance measures at the warehouse. This is measured by the number of mistakes that are made during the picking process. Mistakes can be detected during the process. Right before the boxes are labelled and closed, they pass the incorrect-weight-lane station. Here the boxes are weighed, and the weight is compared to the system-calculated weight. If both weights do not correspond, the box is converted to a station where an operator checks the content of the box. When the content is not correct, the order has to be picked again which results in a lot of rework. The number of mistakes varies on a daily basis. Some days no mistakes are made, other days there are many. The average amount of mistakes detected by the incorrect-weight-lane is 10 per day which is around 0,17% of all pick lines.

If a mistake is not detected during the process, it will end up at the customer. There are three types of mistakes: short shipment (SS), over shipment (OS) and wrong shipment (WS). On average 0,11% of the customer orders in 2020 contained one of these mistakes. Mistakes detected by the customer can lead to customers deciding to order elsewhere and to a lot of rework. New orders have to be created and the products have to be picked and shipped again.

It is of the utmost importance that the method to increase the pick performance resulting from this research will not come at the expense of the quality. It is likely that mistakes can be made more easily when switching to a batch picking process, as several boxes are picked at the same time so products can be placed in the wrong box. Error percentages can be used to compare the quality of the developed method in this research with the current quality. However, this is something that can only be detected once the new method is in place and will therefore not be included in this master thesis project.

2.5 Conclusion Measure phase

This chapter concludes the Measure phase of the DMAIC and gives answers to the first two sub-questions of the research. The first sub-question: "*How can the current process in the repack area be described?"* can be answered as follows: After orders are released by the prep department, the conveyor creates and transport the boxes in where orders have to be picked. The boxes are diverted to the zone where the products are located for that box. An operator scans a box and picks the product. He walks back to the box, places the product in the box and pushes the box back on the conveyor. If all the pick lines of the box are finished, it is transported to the end of the conveyor where the boxes are closed, labelled, and placed in the right truck, ready to be transported by a 3PL provider. Currently, there is no batching in the warehouse. They did divide the repack area into zones and the items are stored with the ABC storage policy. As there is no batching, there is also no proper routing policy in place. The picking process can be described as a low-level, manual, picker-to-parts warehouse

The second sub-question: *"What is the current performance picking process in the repack area?"*, can be answered with the KPI that is used for this research: *the average daily pick line productivity per picking hour.* The average daily pick line productivity per hour was 88,5 pick lines in 2020.

3. Analyse: Order batching in literature

Chapter 3 and 4 form the Analyse phase of the DMAIC. The goal of the Analyse phase is to analyse the data that is collected in the Measure phase. To be able to answer the third sub question: *'What are suitable batch and cluster methods to improve the pick performance in the repack area?',* an overview is given in this chapter of the current scientific literature regarding order batching. First a definition and the general formulation of the problem is given in Section 3.1. Then, common policies to solve the problem are discussed in Section 3.2. Finally, in Section 3.3 the advantages and disadvantages of each policy are discussed.

3.1 Definition and problem formulation

Order batching is the partitioning of the total set of orders into subsets (batches). Multiple orders that belong to the same batch can be picked at the same time and therefore reduce the average travel time. The picked SKUs needs to be sorted when a batching policy is used. This sorting can either be done while picking, sort-while-pick, or after picking, sort-after-pick (Kübler et al., 2020). In sort-while-pick the picked products are directly sorted per customer order. In sort-after-pick, several different customer orders are picked together in the same box or tote and sorted per customer after the products are picked. The sort-while-pick method is most used in practice and researched in literature. Furthermore, this is the current process in the warehouse of the case study, therefore the focus will only be on sort-while-pick batching.

Cergibozan & Tasan (2019) provide a very basic notation of the Order Batching Problem (OBP) in mathematical terms: Assume the set of customer orders is $J = \{1, \ldots, n\}$. C is the capacity of the picking cart (or other device used for picking the batches) and c_j is the capacity required for order j $(j \in J)$. For every batch a vector describes the customer orders: $a_i = (a_{i1}, \ldots, a_{in})$. Binary entries a_{ij} state whether an order is included in a batch $i(a_{ij} = 1)$ or not included $(a_{ij} = 0)$. As the capacity of the picking device cannot be violated by the set of feasible batches, I, the following property holds:

$$\sum_{j \in I} c_j a_{ij} \leq C, \qquad \forall i \in I \tag{1}$$

Binary decision variables x_i $(i \in I)$ are used to describe if the feasible batch is chosen *(x_i =1)* or not *(x_i =0).* The length of the pick tour belonging to batch *I* is denoted with d_i. This leads to the following model:

$$\min \sum_{i \in I} d_i x_i \tag{2}$$

$$subject\ to:$$

$$\sum_{i \in I} a_{ij} x_i = 1, \qquad \forall j \in J \tag{3}$$

$$x_i \in \{0,1\}, \qquad \forall i \in I \tag{4}$$

The objective function (2) will minimize the total length of the pick tour of all the batches. The constraints (3 and 4) make sure that all the customer orders are included in one and only one of the batches. Note that this is a very basic notation of the problem and different studies may use different objective functions and/or constraints to solve the OBP.

Solving the OBP is approached in the literature in two different ways. The first way is to solve the off-line batching problem, where all the orders are known in advance. The second way is the on-line batching problem, where orders arrive during the day and batches are formed based on the known orders (Chen et al., 2018). Even though, the off-line OBP has gotten most attention in literature, the on-line OBP is more realistic (Zhang et al., 2017) because customer orders arrive usually during the day. The on-line batching problem is solved by Zhang et al. (2017) by splitting the on-line problem into smaller off-line problems. They divided the on-line problem into off-line problems based on a fixed timeframe, fixed time window batching (FTWB).

The OBP, is known to be NP-hard (Cergibozan & Tasan, 2019; Kübler et al., 2020). Therefore, there are no optimal solutions found for a large set of orders in established literature and the problem is usually solved by heuristics. The most common heuristics to solve the OBP are further discussed in the next section.

3.2 Order batching policies

According to the literature order batching policies can be divided into four main categories: Priority rules, Seed algorithms, Savings algorithms, and Metaheuristics (Alipour et al., 2020; Scholz & Wäscher, 2017; Zhang et al., 2017). Each of the categories will be discussed briefly.

Priority rules

Priority rules assign orders to batches according to a priority rule. All orders are sequenced based on their priority and allocated to batches according to the sequence, while respecting the limit of the picking device (Scholz & Wäscher, 2017). Most commonly the First In First Out (FIFO) rule is used. This method creates batches based on the arrival times of the orders. This method is often used as a baseline to test the performance of other order batching policies, as FIFO generally does not decrease the average batch tour length (Gibson & Sharp, 1992). Another priority rule is the earliest due date (EDD). This rule groups orders based on their due dates. Orders with early due dates will be processed first and therefore the chance for late orders will be diminished.

Seed algorithms

In seed algorithms first an order is assigned to a new batch based on a seed selection rule. This initial order is called the ‘seed’. An example of a seed selection rule is the Longest Travel Distance. With this rule the order that requires the longest travel distance from the I/O point is chosen as seed for the batch (De Koster et al., 1999). There are two different methods to apply seed selection rules. A single mode, where the seed order is selected a single time. Or a cumulative mode, where the seed is updated every time a new order is added to the batch. The new seed will be calculated from all the orders in the batch (de Koster et al., 1999). Next, unassigned orders are added to this batch based on an order-congruency rule until the capacity of the batch is reached. An example of an order-congruency rule is to add orders that minimize the number of additional aisles that need to be visited (Ho & Tseng, 2006). A seed algorithm creates batches sequentially (Alipour et al., 2020). There are many different seed selection and order-congruency rules that can be used to create batches.

De Koster et al. (1999), Ho & Tseng (2006) and Ho et al. (2008) researched different combinations of seed selection and order-congruency rules. De Koster et al. (1999) mainly focused on seed selection rules based on the aisle of a pick and the travel time. Ho & Tseng (2006) developed seed selection rules based on the specific location of a pick and rules based on the aisle of picks. Ho et al. (2008) extended the research of Ho & Tseng (2006) by also developing rules based on the area all the picks in the batch covered and rules based on the distance between picks. In all three studies, the rules are calculated based on the potential seed order and the I/O point of the warehouse (for the distance and location-based rules) or on the distance between the pick lines within an order (area and aisle-based rules). Ho et al. (2008) found that the distance-based rules performed not very well. They found that rules that considered the smallest number of picking locations, smallest number of picking aisles or smallest rectangular-covering area of an order worked best. The research of Ho & Tseng (2006) also identified the smallest number of picking locations and the smallest number of aisles as the best performing seed selection rules.

In all three papers, the best performing order-congruency rule aims to minimize the additional number of aisles that need to be visited. This is an aisle-based rule. Ho et al. (2008) also proposed some distance-based order-congruency rules. These rules add the next order to the batch based on minimizing the sum of the rectangular or Euclidean distance. There was no statistical difference between these rules and the additional number of aisles rule.

Savings algorithms

According to (Scholz & Wäscher, 2017) the savings algorithm can be defined as follows: for every two orders the savings are calculated (in terms of travel distance) of combining the two orders in one tour instead of having two separate tours for each order. Then the orders which give the most savings as combination are combined in one batch. For savings algorithms, the used routing policy affects the result of the algorithm. The best-known version of the savings algorithm is the Clarke-and-Wright Algorithm (C&W). This algorithm has three variants (De Koster et al., 1999). Other savings algorithms are the EQUAL algorithm and the Small-Large algorithm.

C&W(i)

C&W(i) is the basic variant of the C&W algorithm introduced by Clarke and Wright in 1964. According to de Koster et al. (1999) the algorithm consists of several steps. First (1), the savings for each order pair are calculated, if the pair does not violate the capacity constraint. Then the savings are sorted in a decreasing sequence (2). The pair with the highest savings is selected (3) and added as initial orders to an empty batch if both orders have not been assigned to a batch yet (4). If one of the orders is already assigned to a batch and there is enough capacity for the other order to join this batch, it will be added to the batch. If this is not possible or if the orders are already assigned to two different batches, the next combination from the list is chosen (5). Then the next highest savings pair is selected and the process of step 4 and 5 is continued until all orders are assigned to a batch.

C&W(ii)

C&W(ii) is a variant of C&W(i). In C&W(ii) the savings values are recomputed after orders have been paired together in a batch (Alipour et al., 2020). The algorithm is similar to C&W(i), only steps 4 and 5 are different (De Koster et al., 1999). In step 4 the clusters are combined if the capacity is not violated. If clustering the orders would violate the capacity, the next cluster is chosen. Step 5 checks if all orders are included in a route. If this is the case the algorithm is finished. If not, step 1 is started again. This method will take a lot more calculation time as the savings matrix has to be recalculated several times.

C&W(iii)

According to De Koster et al. (1999) the main downside of C&W(i) and C&W(ii) is that there is no maximum on the number of batches that can be created. C&W(iii) solves this problem. C&W(iii) is an adaptation of C&W(i). From the savings of combining two orders that have not been allocated to a batch yet, a constant c is subtracted. This c can be interpreted as a penalty cost for creating a new batch, which prevents creating additional batches too quickly. This adaptation is simple to carry out without much additional computing time. However, it is difficult to determine what a good value of c would be.

EQUAL

The EQUAL algorithm was introduced by Elsayed and Unal (1989) and can be seen as a combination between a seed and a savings algorithm. It creates batches according to the seed algorithm principle as the combination of the two orders with the highest saving is chosen as the seed. Orders with the largest potential saving in travel time are added to the seed until the batch is full (De Koster et al., 1999).

Small-Large

The Small-Large algorithm from Elsayed and Unal (1989) is discussed in de Koster et al. (1999) makes a distinction between small and large orders. The large orders are batched according to the EQUAL algorithm. The small orders are sorted in decreasing order. The first small order (so the largest order of the small orders) is added to the batch that results in the largest time saving (if possible within the capacity limits). If this order does not fit in any existing batch, a new batch is created (de Koster et al., 1999).

Metaheuristics

Metaheuristics are used to solve the batching problem as they are able to find a good solution for complex mathematical programming problems within reasonable computing time (van Gils et al., 2018). There are many metaheuristics that are used in literature to solve the OBP. The following metaheuristics are examples of metaheuristics used to solve the order batching problem: Genetic Algorithms (Cergibozan & Tasan, 2020), Particle Swarm Optimization (Kübler et al., 2020), Variable Neighbourhood Descend (Scholz et al., 2017) and Iterated Local Search (Alipour et al., 2020; Scholz & Wäscher, 2017)

3.3 Performance order batching policies

In this section, the performance of the four order batching policies is discussed and a conclusion regarding the order batching policies is given.

Priority rules

In general, priority rules are used to compare the performance of other batching policies. Batching orders with a good policy can substantially decrease travel times compared to the FIFO priority rule (de Koster et al., 1999). Especially when batching more than 60 customer orders FIFO performs less than other policies (Alipour et al., 2020). The reason priority rules perform so poorly is that they do not consider the location of the SKUs when creating batches (van Gils et al., 2019). The advantage of using priority rules is that they are one of the easiest methods to implement and require little computation time.

Seed algorithms

The performance of seed algorithms largely depends on the chosen seed selection and order-congruency rule. In general, order selection rules that apply criteria as 'largest, longest or farthest' outperform rules that apply 'smallest, shortest and nearest' criteria (de Koster et al., 1999). Order-congruency rules that work well try to minimize the number of aisles that need to be visited or try to minimize the travel distance by selecting the next order that has the shortest (average) travel distance from the seed order. Furthermore, seed algorithms work in general very well if there are more than 40 orders that need to be batched (de Koster et al., 1999). Besides the good performance, seed algorithms that are based on the single mode do not require a lot of computation time and are easy to implement. Seed algorithms based on the cumulative mode require more computational time, but generally do not lead to better results as the single mode (de Koster et al., 1999). Therefore, the single mode is preferred.

Savings algorithms

De Koster et al. (1999) concluded that in practice the use of the basic variant of C&W, C&W(i), should be sufficient. C&W(ii) and C&W(iii) can outperform C&W(i), but this will take a substantial amount of computational time. The difficulty with the Small-Large algorithm is to determine which orders are large orders. In de Koster et al. (1999) they used the rule that 3/4 of the orders are large and the remaining 1/4 as small. The EQUAL algorithm is always outperformed by the C&W(i) in their research.

Metaheuristics

In general, metaheuristics outperform other batching policies in terms of reducing pick time. For example, Hsu et al. (2005) compared the performance of the Genetic Algorithm with FIFO. They found that the Genetic Algorithm leads to a significant improvement compared to FIFO. However, the metaheuristic methods require long computational times. Especially for situations with many orders. For example, the Variable Neighbourhood Descent method suggested by Scholz et al. (2017) takes up to an hour when having to batch 200 orders. This makes them less suitable for solving the on-line batching problem. Furthermore, metaheuristics can be very complex to understand as well as to implement in practice. It is questionable if the improved solutions outweigh the complexity and calculation time of the metaheuristics.

Conclusion

To conclude, priority rules and seed heuristics perform better in terms of calculation time. However, metaheuristics perform best in reducing pick time. The downside of metaheuristics is that they require long computational times, making them, for the time being, unsuited to solve the on-line batching problem. Large computational times are also a problem for the savings heuristics. The performance of seed heuristics depends on the chosen seed selection rule and order congruency rule. In general, they lead to a good performance if rules are based on minimizing the distance or the number of aisles to be visited. A priority rule like FIFO is generally used to set a baseline to be able to compare the performance of the other algorithms.

4. Analyse: Model

In order to solve the Order Batching Problem at the warehouse, first the conceptual model of the order batching problem for the warehouse is given in Section 4.1. Next, four algorithms have been developed. The first algorithm is based on a combination of different priority rules and the other three are a combination of a priority rule and seed algorithm. All these algorithms are based on the current literature in the field of order batching, as discussed in Chapter 3. In Section 4.2 the four algorithms will be discussed more in depth. An approach to solve the on-line problem by splitting it into smaller off-line problems is discussed in Section 4.3. Next, in Section 4.4, the input parameters of the model are discussed and the corresponding assumptions in Section 4.5. Finally, the models are validated and verified in Section 4.6 and different implementation options are proposed in Section 4.7.

4.1 Mathematical formulation

In this section, a conceptual model of the batching problem at the warehouse is given. The model of Lolkema (2020) was adjusted to fit the specific problem characteristics of the company.

In this model it is assumed that only boxes with a single line order are picked. This means that each box only has one location visit. The pick locations and corresponding zones are determined beforehand. Furthermore, it is assumed that all the orders can be fulfilled with the products on stock. Hence, there are no replenishments needed. The warehouse is divided into zones and each pick cart can only pick products from the same zone.

Orders are directly picked in the shipping box. These boxes are placed on the picking cart. The capacity on the cart therefore depends on the number of boxes that fit on a cart and not the number of pick lines or locations that have to be visited. The boxes only contain small and light products; therefore, no further restrictions on the volume and weight are needed. Furthermore, a distinction can be made between the priority of boxes. They can have a high priority, number 1, or a low priority, number 5. The model has the following input variables and decision variables:

Input variables

β_0	Fixed set – up time per batch
β_1	Variable set – up time per box in a batch
β_2	Set – up time single order pick
β_3	Travel time between locations batch pick
β_4	Travel time to and from I/O point batch pick
β_5	Travel time to and from I/O point single order pick
$O = \{1, \dots, \lvert O \rvert\}$	Set of boxes
$M = \{1, \dots, \lvert M \rvert\}$	Set of pick lines
$B = \{1, \dots, \lvert B \rvert\}$	Set of batches
C	Maximum capacity of pick cart
$d_{I/O,i}$	Distance between the I/O point and storage location i

d_{ij}	Distance between storage location i and j
$p_0 = \{1,5\}$	Priority order o, where $o \subseteq O$

Decision variables

$$A_o^b = \begin{cases} 1, \text{if box } o \text{ is assigned to batch } b \\ 0, \text{otherwise} \end{cases}$$

$$D_b = \begin{cases} 1, \text{if batch } b \text{ has } n_b > 1 \\ 0, \text{otherwise} \end{cases}$$

$$E_b = \begin{cases} 1, \text{if batch } b \text{ has } n_b = 1 \\ 0, \text{otherwise} \end{cases}$$

In this research, boxes can either be picked in a batch, or they are picked by the single order picking policy. If boxes are batched, they need to be placed on a pick cart. Placing the boxes on a cart takes time. This time can be divided in a fixed set-up time (β_0), and a variable set-up time (β_1), depending on the number of boxes in the batch. If the box is not batched, and picked by the single order picking policy, there is also a set-up time. However, this set-up time is different from the batch set-up time and therefore denoted by β_2. These set-up times are measured in seconds and further explained in Section 4.4.

Travel distances are measured in seconds. There are two types of travel distance: from one pick location to another pick location, and from the I/O point to the first pick location or from the last location back to the I/O point. If a batch is picked, an operator has to walk with a pick cart which will result in a longer travel time. The travel time between locations is denoted as β_3 and from the I/O point to the first pick location or from the last location to the I/O point as β_4. When the single picking policy is used, an operator only needs to travel to and from the I/O point to the pick location. This travel distance will also be faster than the batch picking travel time as the operator can walk without a pick cart. This travel time is denoted by β_5 and is in seconds.

The distances between pick locations, and between pick locations and the I/O point are calculated by Dijkstra's Algorithm. With this algorithm the distance d_{ij} between location i and location j and the distance $d_{I/O,i}$ between the I/O point and start or end location i can be calculated. Only distances between locations in the same zone can be calculated.

The number of boxes in a batch can be determined with the following formula:

$$n_b = \sum_{o \in O} A_o^b \tag{5}$$

If the number of boxes is higher than 1, batches are formed and the time it takes to pick a batch is calculated. If the number of boxes is equal to one, orders will be picked as a single order pick and the pick time is calculated accordingly.

Boxes are assigned to a specific batch: $\{A_o^b\}_{o \subseteq O}$. After the batches are formed, the batch route with the sequence of the storage locations to be visited can be determined by the WMS. The sequence of locations to visit on a pick tour is denoted $(i1, \dots, in)$. The pick time per batch can now be calculated with Formula 6.

$$T\left(b,\{A_o^b\}_{o\subseteq O}\right) = t_b^{set-up} + t_b^{travel} \tag{6}$$

The goal of the model is to minimize the total pick time of all the batches in the warehouse. This will be done with the following objective function:

Objective

$$minimize \sum_{b\subseteq B} T(b,\{A_o^b\}_{o\subseteq O}) \tag{7}$$

Subject to

$$\sum_{o\subseteq O} A_o^b \leq C, \qquad \forall b \subseteq B \tag{8}$$

$$\sum_{b\subseteq B} A_o^b = 1, \qquad \forall o \subseteq O \tag{9}$$

$$t^{set-up} = D_b \cdot \left(\beta_0 + \beta_1 \cdot n_b\right) + E_b \cdot \beta_3 \tag{10}$$

$$\begin{aligned} t^{travel} = D_b \cdot (\beta_3 \cdot \left(d_{i1,i2} + d_{i2,i3} + \cdots + d_{i_{n_{b-1}},i_{n_b}}\right) + \beta_4 \cdot \left(d_{I/O,i1} + d_{i_{n_b},I/O}\right)) \\ +\ E_b \cdot (\beta_5 \cdot \left(d_{I/O,i1} + d_{i_{n_b},I/O}\right)) \end{aligned} \tag{11}$$

$$A_o^b, D_b, E_b \subseteq \{0,1\}, \quad \forall o \subseteq O, b \subseteq B \tag{12}$$

The objective function (Formula 7) is to minimize the sum of the time it takes to pick each batch. To solve this objective function there are several constraints. Constraint 8 makes sure that each batch does not have more boxes than the batch capacity allows. Constraint 9 states that each box is assigned to only one batch. Constraint 10 calculates the set-up time for each batch and Constraint 11 the travel time per batch. Lastly, Constraint 12 ensures that each decision variable has either a 0 or a 1.

As mentioned in Chapter 3, the OBP is known to be NP-hard. This means that no optimal solutions can be found for a large set of orders and the problem is usually solved by heuristics. The model presented in this section will also be solved by heuristics. These will be discussed more in-depth in the next section, Section 4.2.

4.2 Algorithms

As mentioned in Section 4.1, no optimal solution can be found for the model. Therefore, it is solved by heuristics. Four algorithms are created using priority rules and seed heuristics. The reason to only focus on these two batching policies and not on the other two policies discussed in Chapter 3, savings heuristics and metaheuristics, is that metaheuristics add a lot of complexity while it is questionable if their improved solution can outweigh this complexity. Furthermore, these are harder to implement in practice and require more computational time, which is not convenient as the on-line batching problem requires several off-line problems to be solved during the day. Large computational time is also the reason that savings heuristics are not considered. The research of de Koster et al. (1999) suggests using seed heuristics when there are more than 40 orders that require batching, otherwise the computational times become very large. Therefore, this research focuses on seed heuristics and uses a priority rule algorithm as a baseline to get an indication of the quality of the proposed seed heuristics.

Three algorithms are based on the top-performing heuristics proposed in the research of de Koster et al. (1999), Ho & Tseng (2006) and Ho et al. (2008), that are suited for the warehouse characteristics of the company. This resulted in one seed selection rule: the box that is furthest away from the I/O point, and two order-congruency rules: add the box that results in the smallest number of additional aisles to be visited (1) and the box with the shortest distance in seconds with respect to the seed order (2). The heuristics are slightly adapted, so they are able to take into account the different priorities of orders in the warehouse. The first algorithm is based on the priority rule FIFO, which is commonly used in literature to benchmark the performance of the other algorithms. In total, four algorithms are proposed, each of them will be discussed shortly.

4.2.1 Algorithm 1: EDD - FIFO

The first algorithm uses a combination of the priority rules Earliest Due Date (EDD) and First In First Out (FIFO). As mentioned in Section 2.2.2 a box can either belong to one of the two priority groups. Boxes of the first priority group have an earlier due date and have to be picked before the boxes of the second priority group. Boxes within a priority group are batched on the FIFO principle. Therefore, boxes are first grouped by priority and then batched based on the FIFO principle. The main advantage of this algorithm is that it is easy to implement in the current process, as operators already know which boxes have priority. Therefore, they will only have to place a number of boxes, equal to the batch capacity, on the picking device. Moreover, this batching algorithm resembles the current process, where boxes with a higher priority are picked first in a first-in first-out sequence.

Moreover, this algorithm can be used to model the current situation which is equal to a batch capacity of one box, as currently there is a single order picking policy in place. The performance of the current situation will be used as a benchmark performance for all the batching algorithms to see if batching orders improves the performance of the conveyor area.

The FIFO rule is commonly used in literature to assess the performance of batching heuristics, as mentioned in Section 3.2. Therefore, the EDD-FIFO rule for a batch capacity higher than 1 can be used as a base line to see if more sophisticated batching methods lead to a higher performance than randomly creating batches.

4.2.2 Algorithm 2 - Seed heuristic smallest number of aisles

This algorithm is based on a combination of a seed heuristic and priority rule. From the boxes with the highest priority still available in the order pool, the box with the furthest location in terms of row number and distance to the conveyor within the row, is chosen as seed. This seed selection rule can be seen as an adaptation of the rule of de Koster et al. (1999) The order-congruency rule is adapted from Ho & Tseng (2006) and Ho et al. (2008). They proposed to add the box to the batch that results in the smallest number of additional aisles that have to be visited. Therefore, first boxes of the same aisle are added to the batch and if there is still space, boxes of the adjacent aisle are added. This continues until the batch is full. This rule takes the priority into account as well. First, the boxes from the highest priority group still available in the order pool are added. If there are no more boxes from this group, the next priority group is considered.

4.2.3 Algorithm 3 - Seed heuristic with distance

The third algorithm is a combination of a seed heuristic with a priority rule. From the available boxes in the order pool, the one with the highest priority is chosen as seed order. If there are more boxes with the same priority, the box with the furthest distance (measured in travel time) from the I/O point is chosen as seed. This seed selection rule is an adaptation of the rule from de Koster et al. (1999), to pick the order with the longest travel time as seed order. The order-congruency rule adds the orders with the shortest distance in seconds with respect to the seed order, to the batch until the capacity of the batch is reached. According to de Koster (1999) and Ho et al. (2008) a rule that focuses on minimizing the sum of distances performs very well. This rule is similar to the one in Algorithm 2. The difference is that the distance in Algorithm 2 is based on the row and bay number, where the distance in Algorithm 3 is based on the travel distance in seconds. Also, for the order-congruency rule the priority is considered. This means that in the pool of available boxes, first the group with the highest priority is added to the batch. If there are no more boxes in this group, the next priority group is considered.

4.2.4 Algorithm 4 - Seed heuristic same aisle

This algorithm is similar to Algorithm 2, only this algorithm exclusively allows boxes with the same aisle as the seed to be added to the batch. Therefore, an operator will only need to visit one aisle to complete picking this batch. This is an extreme version of the smallest number of additional aisle rule of Ho & Tseng (2006) and Ho et al. (2008), which resulted in the best performance in the respective papers. However, due to the “only same aisle” restriction, it is possible that the batches are not filled to their full capacity. This could significantly decrease the performance of this algorithm, especially for higher batch capacities.

Each algorithm is tested for different settings concerning the capacity of the pick cart and which bays in the warehouse are allowed to be batched. Currently, there is no batching, so it is important to know if and how well each method works for a certain batch capacity. The company has a storage location assignment based on the ABC logic. Therefore, it might be more beneficial to only batch the slow-movers located at the end of an aisle, as the fast-movers are so close to the conveyor, the set-up times for creating a batch might be larger than walking to these products.

4.2.5 Example algorithms

To make clear how the designed algorithms work, the warehouse map in Figure 6 is drawn with pick lines and the corresponding formed batches per algorithm. In this warehouse several pick lines have to be picked. Each pick line corresponds to one box. These pick lines are numbered 1 to 6. The number of the pick line corresponds to the time the box entered the warehouse, i.d. pick line 1 was the first box to be available and pick line 6 the last. For each algorithm it is demonstrated which pick lines will be in the same batch, if the batch capacity equals three pick boxes.

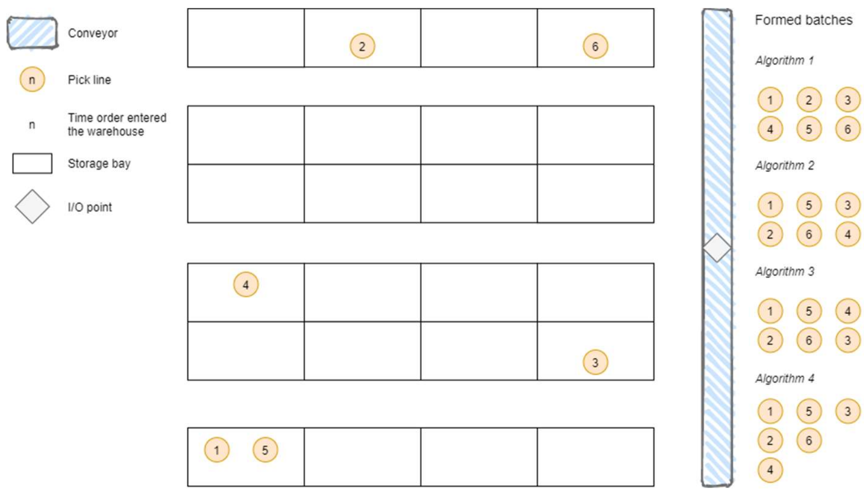


Figure 6: Warehouse map with pick lines and batches for a capacity of three boxes

Algorithm 1: EDD – FIFO

For this algorithm, the only things that are considered when forming the batches is the priority (which is not included in this example) and the time the pick lines entered the warehouse. Batches are made with the first-come first-serve principle. Therefore, two batches are formed, and these batches consist of the pick lines: (1, 2, 3) and (4, 5, 6).

Algorithm 2: Seed – Aisle

For this algorithm, the order that is furthest away from the I/O point (the diamond on the conveyor in Figure 6) is chosen as the seed for the batch. This is order 1. Pick lines are added based on the aisle number of the locations of the pick lines. Pick line 5 and 3 are in the same aisle, so they are added to the batch. Now the capacity of 3 is reached, so a new seed can be chosen. Of the pick lines that are left (2, 4 and 6), pick line 2 is located furthest away from the I/O point. This order will become the new seed. Next, pick line 6 is added to the batch as it is in the same aisle as pick line 2. As the capacity of 3 is not yet reached, the next aisle is considered. The next aisle has pick line 4, and this pick line is added to the batch. In total, two batches are formed, consisting of pick lines (1, 5, 3) and (2, 6, 4).

Algorithm 3: Seed – Distance

This algorithm also starts with the pick line that is furthest away from the I/O point, pick line 1. This pick line will be the seed. Pick lines are added based on the distance from this seed. Therefore, pick line 5 and 4 are added to this batch as they are closest to pick line 1. The first batch is complete, and the next seed order can be chosen. This will be pick line 2 and the remaining pick lines, pick line 3 and 6 are added to complete this batch. The resulting two batches are pick lines (1, 5, 4) and (2, 6, 3).

Algorithm 4: Seed – Same aisle

This algorithm is similar to Algorithm 2, with the difference that pick lines can only be in the same aisle as the seed order. Therefore, this algorithm will generally result in more batches than Algorithm 2. This Algorithm will result in 3 batches, with pick lines (1, 5, 3), (2, 6) and (4).

4.3 Timeframes

The warehouse receives the orders on the same day as they have to be processed. This makes it impossible to solve the batching problem off-line (dividing the orders in batches when all the orders are known in advance) and the problem has to be solved on-line. Zhang et al. (2017) and Alipour et al. (2020) proposed to solve the on-line problem by splitting the on-line problem into smaller off-line problems. These off-line problems can then be solved to find a solution to the overarching on-line problem.

In this research, the approach of Zhang et al. (2017) is applied to create off-line problems. The researchers propose to divide the on-line problem into smaller off-line problems based on fixed timeframes, fixed time window batching (FTWB). For this approach, fixed timeframes have to be defined and all the orders that enter the system within this timeframe will be collected and batched after the timeframe has passed. A disadvantage of this method is that it is not suitable in a fluctuating order environment, where the size of collected orders in a given timeframe can vary greatly per day. In Section 2.4.1 the time of the orders entering the system were analysed and it was concluded that even though there is a fluctuation in the arrival of orders on a daily basis, there is no fluctuation on a day-to-day basis. Therefore, fixed timeframe batching will be used in this research to create the smaller off-line problems.

The repack area starts working at 13:00h. This means that at 12:30h the latest, orders have to be released to ensure there is enough workload for the operators. The official deadline to order is at 18:30h. There are some customers that are allowed to order after this cut off time. However, the number of orders that arrive after 18:30h is very low and therefore it is assumed that the last order arrives at 18:30h. After consulting with the company, four fixed timeframes are proposed:

1. All orders that entered the system before 12:30h.
2. Orders that entered the system between 12:30h and 14:00h.
3. Orders that entered the system between 14:00h and 16:30h.
4. Orders that entered the system after 16:30h.

These timeframes have also been tested by asking the prep department to wait with releasing the boxes as long as possible. To see the influence of the timeframes on the performance of the algorithms, different timeframes are proposed in Chapter 6.

All orders that arrive within a timeframe will be saved up and released at the end of the respective timeframe. If the orders released in the specific timeframe cannot be picked in that same timeframe, they will be picked in the beginning of the next timeframe. The batches will remain the same, so these orders will not be re-batched with the orders from the next timeframe. The reason for this is that once orders are released in the system, they cannot be unreleased and assigned to a different batch.

4.4 Input parameters

This section will briefly discuss the input parameters for the four models. All models have the same input parameters. Except for the timeframes, this is only an input parameter for algorithms 2, 3 and 4.

Timeframes

The timeframes are used to create smaller off-line problems of the on-line batching problem at the warehouse. As mentioned in Section 4.3, there are four timeframes: before 12:30h, between 12:30h and 14:00h, between 14:00h and 16:30h, and after 16:30h. Only for the Algorithm 1 the timeframes are not used. This is because the algorithm creates batches on the FIFO principle, so there is no need to divide the orders into separate groups.

Historical order data on pick line level

With the historical pick line data, it can be determined how many pick lines were intended for the repack area. Furthermore, the exact time the order entered the ERP system can be determined with this data. The locations of the SKU, the priority of the pick line, box size and box number can be retrieved from this data as well. The input data for the model is the daily pick line data for 12 representative days in February and March of 2021. The days can be divided into three categories: days with less picks than average (‘Quiet’ days), days with average picks (‘Normal’ days), and days with more picks than average (‘Busy’ days). On average the number of pick lines of all 12 days is 5.799, the minimum 4.688 and the maximum 7.443 pick lines.

Travel times

To be able to determine the travel times in the repack area, a time study was conducted in the warehouse. In the warehouse there are three types of possible movements: 1) an operator walks from the conveyor to the middle of the first bay or from the middle of the last bay to the cross aisle. 2) from the middle of any given bay to the middle of the previous or next bay. 3) from the front or back of one aisle to the front or back of the previous or next aisle. In the time study two different persons walked these three distances multiple times to determine the average travel time for each movement.

Set-up time batching

The set-up time when batching orders consists of two components, a fixed and variable component. The fixed component consists of the steps ‘Scan cart’, ‘Confirm batch’ and ‘Remove boxes from cart’ (for all the steps of the batching process, see Appendix C). This set up time is the same for all batches, irrespective of the number of boxes on the cart (unless there is only 1 box to be picked, in this case it will always be assumed the order is picked by a single order picking policy). The variable component does depend on the number of boxes in the batch and consists of the steps ‘Scan position’, ‘Scan box’ and ‘Place box on cart’. These set up times (fixed and variable) are also determined by a time study where two different persons performed the steps several times.

Set up times single picking policy

In the case there is no batching implemented (batch capacity = 1), there are still set up times in the process. Boxes must be scanned before the pick can start and pushed back on the conveyor belt after the pick is completed. These set up times are retrieved from the VSM in Section 2.4.2 (Figure 5).

Batch capacity

As there is no batching yet, it is important to know for which cart capacity it will be feasible to batch orders. Furthermore, it is important to know which capacities are feasible concerning the layout of the warehouse. This has been measured extensively and verified by a company selling pick carts. The most feasible capacities for the warehouse are 9 or 12 boxes. Therefore, most simulations in Chapter 5 will be run for these batch capacities.

Storage bays

Zones in the warehouse have three or four bays. It is interesting to see if it makes sense to batch all the orders from all the bays or if it is better to only batch orders from the bays that are located further from the conveyor. The reasoning for this is that fast-movers are located in the front of the aisles. Therefore, it might take more set-up time to place these orders on a batch, than the actual time it takes to pick them. For each algorithm it will be tested if the best option is to batch all four bays, the last three bays, the last two bays or only the last bay.

4.5 Assumptions

Besides the model inputs, also several assumptions and simplifications are made to be able to model the process. The main assumptions and their implications are discussed in this section.

Picking times

It is assumed that batching only influences three steps of the eight steps of the picking process described in Section 2.4.2 (see Appendix C for a business process diagram of the batch picking process and Appendix B for the diagram of the single order picking process). The steps that are influenced are ‘Walk to location’, ‘Scan box’ and ‘Place box on conveyor’. ‘Walk to location’ changes as an operator does not have to return to the I/O point after picking the orders for a single box. ‘Scan box’ changes as scanning boxes to create batches takes longer than only scanning them for a single pick and the boxes have to be placed on a pick cart with batch picking. Lastly, ‘Place box on conveyor’ will take longer when batching orders, as the boxes must be taken from the pick cart and placed back on the conveyor belt. The other steps will take the same amount of time, regardless if the orders are batched. As mentioned in Section 4.4, the changes in picking times with batching are extensively tested. Therefore, it is likely that this assumption will not change the actual result that much.

Storage locations only on bay level

In this research it is assumed that all the products located in the same bay are in the same place, i.e., the middle of the bay in terms of height as well as width. The height assumption is made as it is assumed that taking products from the top, bottom or middle shelf takes the same amount of time regardless if orders are batched. The width assumption is made for simplicity reasons. Bays with large products only have three bins per shelf, where bays with small products can have as much as ten bins for products. From the location number in the historical data, it is impossible to know where exactly the product is located in terms of width of the storage bay. Therefore, it is assumed that all the products are located in the middle of the bay. This assumption might slightly overestimate the savings for the batching algorithms as products are placed in the aisles with the ABC logic (Section 2.3.2). This means that products with higher demand will be located in the front of the bay and will take slightly less walking time then to the middle of the bay, the time that is taken in this research.

Only batch small boxes

In the repack area, products can be shipped in four types of boxes. Two of these boxes are considered to be too big for batching. Placing these on a cart will take up too much place and these boxes contain, in general, more and heavier products. It will be too heavy for the operators to lift these kinds of boxes on and from a cart multiple times a day. Therefore, only the two small box types will be considered for batching and the other boxes are picked as if the batch capacity equals one (so as the current single order picking policy). If the company decides to implement batching only for the small boxes, this assumption will have no impact on the outcome of the research.

Location I/O point

It is assumed that every zone has its own I/O point to pick up and drop off boxes in the middle of the conveyor belt. In reality, operators can place boxes on the conveyor at every point of the belt. However, for simplicity reasons it is assumed this can only be done in the middle. Although this assumption will overestimate the time it takes for an operator to return to the conveyor, the operator still has to take new boxes somewhere from the conveyor. It is likely that these times cancel each other out and that this assumption hardly impacts the results of the research.

Fixed work pace operators

It is assumed that all operators work at the same pace and keep this pace throughout their workday. This is not realistic in practice, as people will experience fatigue at the end of the workday, or one operator will work faster than another. Moreover, the work pace is assumed to be independent on the amount of work on a day. A fixed work pace is a common assumption in literature as accounting for different paces is very difficult. Furthermore, the current performance is also modelled with this assumption in mind. Therefore, the effect on the performance of the algorithms with this assumption is negligible.

Travel times single order pick and batch pick

It is assumed that travel times when picking a single pick order (without a cart) are faster than walking with a cart. It is difficult to determine how much more time picking with a cart will take as there are no picking carts yet and the capacity of the cart is unknown. To get an idea of the travel times with a cart, a cart with realistic dimensions for an order pick cart is used to walk through several aisles of the conveyor area. The same distance is travelled without the cart. The difference in percentage between these times is calculated and, when the batch capacity is higher than one, added as a mark-up to the total travel time of the distance for a batch capacity of one. This assumption might have some influence on the results. If the travel times with a cart are lower than estimated in this research, the results will be more favourable towards batching orders. However, if the walking times are higher, the saving of batching might be less than anticipated.

No difference in travel time for a different batch capacity

No distinction is made between walking times for varying batch capacities. It is assumed that picking orders for a capacity of 3 takes as much additional walking and searching time as a capacity of 30. This assumption is not very realistic, as for a capacity of 30 boxes it will take an operator more time to find the right box for each order and the cart will be more difficult to handle then a small cart. Therefore, it is likely that this assumption overestimates the savings for higher capacities and underestimates the saving for low capacities. However, the algorithms are tested in this research for the capacities of 9 and 12 boxes (the motivation for these capacities will be explained in Section 5.1). The carts will have a similar size, only the 12-box cart will have one extra shelve. This means that there will be hardly any difference between these two carts in terms of size and weight and therefore, hardly any difference in travel time.

No picker blocking

Usually there is only one operator working in a zone, therefore it is assumed there is no picker blocking in this research. However, even though operators working on the outbound process (order picking) do not block each other, it is possible that there are other inbound operators (stocking the products) working at the same time. They might be hindered by the outbound operators when they are picking the batches. It is expected that this form of blocking does not result in major issues as the inbound team has their main workload in the morning and the outbound team in the afternoon.

No extra preparation time to make batches for the prep department

If orders have to be released in a different way to create the batches, this will likely lead to extra time for the prep department. However, in this research this extra time is neglected. This time is neglected as it is hard to determine beforehand how much time this will take. If the prep department does need a significant amount of time to prepare and release the batches, this time should be included in the model and the savings calculated by the models will decrease.

Bulk locations not considered

The conveyor area has a few bulk locations where a pallet can be stored. Bulk locations are only used if the company knows in advance that they have a lot of orders from a specific product. These locations are very close to the conveyor belt, so the walking time is minimized. Due to the strategic location of the bulk locations and the irregular use of these locations, they are not considered in this research.

Multi-line boxes

At the end of the picking process in a zone, the boxes have to be removed from the picking cart. If all the boxes are completed, the boxes are deleted automatically from the cart in the system and the boxes can be removed without a problem. However, if a box still has pick lines in another zone (a multi-line box), the box has to be manually removed from the cart. Otherwise, it cannot be added to the cart of the operator in the next zone. Ideally, all the boxes can be manually removed from the cart with several simple steps in the RF scanner. However, for this to work a system change is needed, which will be discussed further in Section 4.7. For simplicity reasons it is assumed that all the carts will have this step in the end, not only those who still have pick locations in other zones. The time needed for removing the boxes is included in the fixed set up time of a batch. This is only relevant for the batching all lines implementation option, which will be discussed in Section 4.7. This assumption will likely overestimate the fixed set-up time for a batch when all lines are batched, as there will be batches that only have boxes with pick lines within the same zone.

No machine breakdown

The process in the repack area is dependent on several machines. The boxes are made in the beginning by special box making machines, the boxes have to be transported over the conveyor belt, and at the end the boxes have to be closed and labelled. These machines occasionally break down. In this research, these breakdowns are not considered, and it is assumed the process can run smoothly without interruptions during the day. This assumption can have implications for the results of this research as the machine errors can delay the arrival of boxes needed for a specific batch. Batches cannot be formed as calculated by the models and it is unclear what kind of batches will be formed and what the savings of these batches will be.

Priority orders

As mentioned in Section 2.2.2, all boxes have a priority number on them. The numbers range from 1 to 8 with 1 having the highest priority and 8 the lowest. In consultation with the company, it was decided to decrease the priority groups from eight to two (group 1 being boxes with numbers 1 to 4 and group two boxes with numbers 5 to 8) as firstly, some priority groups are very small or hardly used. Secondly, the deadline for each group does not differ that much from the others they are grouped with, so it will likely not create any problems regarding deadlines. And lastly, creating bigger priority groups results in more possible combinations and a better performance of the algorithms.

No hot replenishments needed

It is assumed that all orders can be released immediately after they enter the system. In reality, this is not always the case as sometimes there is not enough stock in the repack area to fulfil an order. When this happens, a hot replenishment order has to be carried out to restock the products before the order can be released. However, the number of hot replenishments on a day is relatively small and hard to predict beforehand. Therefore, these are not considered in this research. In practice, this assumption can lead to a decrease of the performance as some orders that are assigned to batch X in the models, have to be picked later in batch Y as the order cannot be released before the hot replenishment is done.

4.6 Validation and verification

In order to determine if the proposed simulation models are correct, they have to be validated and verified.

4.6.1 Validation

The models need to be validated to be sure that they resemble the real-world situation. There are four types of validation activities that need to be performed when building a simulation model: conceptual, logical, experimental, operational and data validation. For each of these types, it is briefly argued why the models meet the validation criteria.

The *conceptual validity* of the models was tested by extensively discussing the steps in the models with site manager of the warehouse. The site manager has expert knowledge of all the current processes in the company and is therefore able to validate all the assumptions in the models. *Logical validity* is similar to the concept of verification and will therefore be discussed in Section 4.6.2. *Experimental validity* refers to the quality and efficiency of the solution mechanism and the sensitivity of the solution when changes are made in the models' parameters (Landry et al., 1983). This is tested by the sensitivity analysis in Chapter 6 and the different scenarios proposed in Chapter 7. According to (Sargent, 2010), *operational validity* is achieved when the models produce information to help decision makers to accept or reject a solution. The results of the proposed models provide a comparison with the current system in place and with each other. Therefore, they can be used by decision makers to decide if order batching should be implemented and in which form. Finally, *data validity* concerns amongst others the correctness and appropriateness of the data (Tsioptsias et al., 2016). The data used in the models is the historical daily order data of the warehouse. The data is from 2021 and therefore reflects the current situation in the warehouse. Furthermore, this data is regularly used in other company analyses and decision making, and it can therefore be assumed that it is valid to use for the simulation models.

4.6.2 Verification

Verification is similar to the aforementioned validation type *logical validity* (Section 4.6.1). It is needed to check if the results of the models are accurate and if there are no bugs or errors in the code. All the models are written in Python and are debugged to remove the bugs and errors. To check if the models are accurate, they were built in a step-by-step process. After each step, extensive checks were performed to see if the step worked as intended. Checking the models step by step made it easier to recalculate instances by hand and therefore to detect mistakes. When the models were finished, a test run was conducted, and intermediate results of the test were obtained and compared with calculations by hand.

4.7 Implementation options

Two aspects need to be considered when running the models. The first aspect is if all the lines can be batched or only the single line orders. The second aspect concerns the routing.

All boxes or only single lines

Ideally, all the boxes will be batched. However, with the current settings of the batching program in the new WMS boxes are only removed from a cart when all the pick lines are completed. This means that when an operator finishes all the picks in one zone, he or she cannot remove the box from the cart as it still has picks left in another zone. If this setting remains unchanged, batching can only be done for single line boxes. These boxes only have one pick line; therefore, all the picks will only be in one zone. As it is not sure what the costs of this change will be, all the algorithms are tested for the option where only the single lines are batched and the option in which all the lines can be batched.

Current or new routing policy

The pick system sends the operators to the locations in an alphanumeric order. Currently the locations at the warehouse are numbered in such a way that this leads to very inefficient routes. As can be seen in Figure 7, the bays on one side have the numbers 0 to 3 and the numbers on the other side 5 to 8. This means that a picker has to pick first all the products on one side of the aisle, walk back to the beginning of the aisle, and pick all the orders from the other side. Figure 8 shows a proposed change in the numbering of the warehouse locations. This change will make sure the routing policy resembles the S-Shape heuristic. The S-shape heuristic is defined by de Koster et al. (1999) as follows: all the aisles with requested products are entered and traversed completely.

Changing the routing can be done in two ways: the physical numbers of each location can be changed or the sequence in the system can be altered. Changing the physical numbering will take a lot of time and therefore this will result in substantial costs. Changing only the sequence will be relatively cheap. However, inbound operators, who stock the products in the warehouse, use the same RF program and therefore sequence. The way products are stocked is connected to the sequence. Therefore, changes in the logic at outbound can also influence the performance at inbound. More research is needed to see what the impact of these changes will be on the performance of the inbound department. However, this is not part of the scope of this project. Therefore, this research assumes that all the locations have to be physically renumbered.

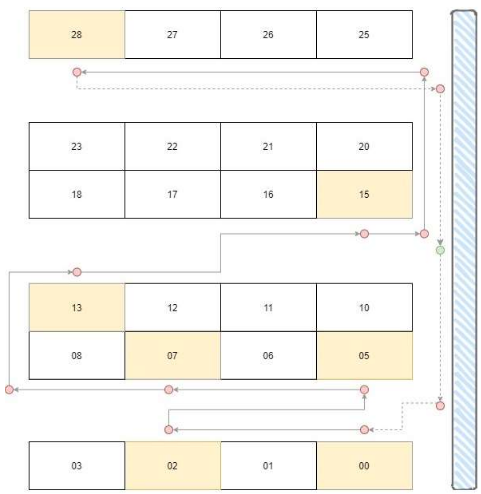


Figure 7: Current numbering

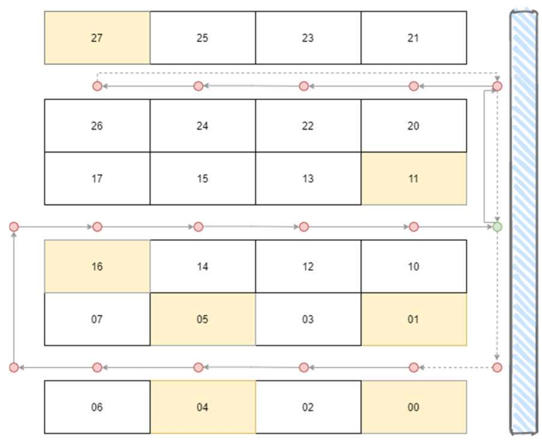


Figure 8: Proposed new numbering

As there are two options in terms of lines to batch (all lines or only single lines) and two routing options, there are four implementation options in total that will be considered. These four options are:

1. Single lines with the current routing policy
2. Single lines with the new routing policy
3. All lines with the current routing policy
4. All lines with the new routing policy.

4.8 Conclusion Analyse phase

Now the third and fourth sub-question can be answered. Sub-question 3 stated: “*What are suitable batch and cluster methods to improve the pick performance in the repack area?”* From the literature in Chapter 3 followed that there are four common types of heuristics used to solve the OBP. These are priority rules, seed heuristics, savings heuristics, and metaheuristics. From these four methods only the priority rules and seed heuristics were considered to be suitable for the repack area. Metaheuristics are not suitable as they add a lot of complexity while they do not lead to much better solutions. Moreover, these are very difficult to implement in practice and take a lot of computational time. Savings heuristics also take a lot of computational time. Therefore, this heuristic is also not suitable for the repack area as several OBPs need to be solved throughout the day and the solutions need to be calculated fast. Priority rules and seed heuristics are considered to be suitable as their computation time is small and they are easy to implement in practice.

To answer the fourth sub-question: *"How can the performance of these methods be calculated?"* four algorithms were developed. The first algorithm uses a combination of priority rules, namely the EDD and the FIFO rule. The other three algorithms are based on seed heuristics. Algorithm 2 makes batches based on the nearest aisles, Algorithm 3 on the closest distance and in Algorithm 4 only pick lines from the same aisle are allowed to be batched. The algorithms are simulated using historical production data of 12 days in February and March 2021. Furthermore, the simulations are run for four different implementation options: batching only single line boxes with the current routing (1), batching only single line boxes with the new routing (2), batching all the lines with the current routing (3) and batching all the lines with the new routing (4). The results of these simulations will be presented in the following chapter, Chapter 5.

5. Improve: Results of the case study

The Improve phase of the DMAIC consists of Chapter 5, 6, 7 and 8. The goal of the Improve phase is to find solutions to the problem and to validate these solutions. In this chapter the results of the simulations will be analysed. Firstly, it is examined in Section 5.1 if batching is beneficial for all types of orders or only orders with slow-moving products. Next, in Section 5.2 it is analysed which batching algorithm works best for the current situation at the warehouse. In Section 5.3 it is reviewed if the performance of this algorithm identified in the previous section depends on the different day types. Furthermore. In Section 5.4 the results are analysed for the proposed new routing policy in Section 4.7. In Section 5.5 the savings are calculated when all the pick lines are allowed to be batched and not only the single lines. Section 5.6 combines the picking all lines option with the new routing policy. Then in Section 5.7 a summary is given of all the four options and finally, in Section 5.8 the findings of this chapter are summarized.

5.1 Identify bays to be batched

The company stores their products according to the ABC logic. This entails that fast movers are located in the front of an aisle, mainly in the first bay, and slow movers are located in the successive bays. An area of interest is to investigate if it is better to batch all the orders or to batch only the slow movers. The idea is that fast movers are located so close to the conveyor that the set-up time exceeds the pick time if these picks are batched.

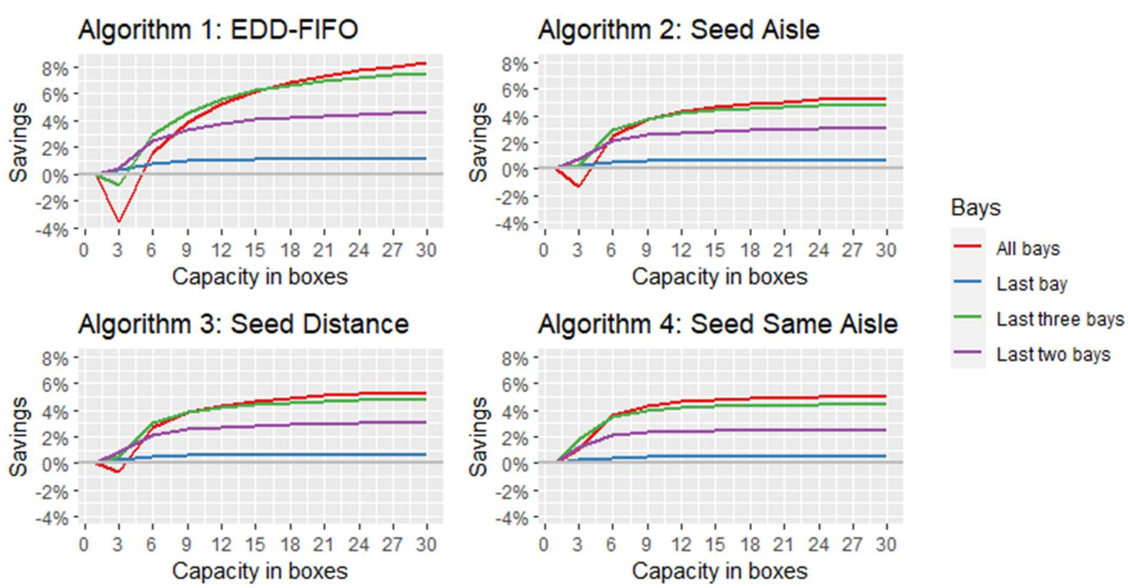


Figure 9: Performance per bay batching option

Figure 9 shows the performance for each bay batching option and algorithm. As can be seen, the graphs for all four algorithms have a similar trajectory for the saving in average walking time per batch capacity for each bay batching option. For low batch capacities it is generally better to focus only on batching the last or the last two bays, so only the slow movers. Visiting these locations will requires more travel time, so the saving in travel time will outweigh the set-up times for a batch, even at low capacities. If the batch capacity increases, the performance of batching only these bays stagnates and it becomes more interesting to batch the last three bays or even all four bays, hence, to batch all the products. The performance of only batching the last or the last two bays stagnates as there are only a limited number of picks in these sections. Batching the last three bays or all the bays becomes more interesting at higher capacities as a higher capacity will increase the pick density and therefore lead to more saving in travel time. For each algorithm the performance per bay batching option will be briefly discussed.

For Algorithm 1 it becomes beneficial to batch all bays if the batch capacity exceeds 15 boxes. Only if the batch capacity is smaller or equal to 4 it is beneficial to batch only the last or the last two bays. However, the percentage saving compared to a single picking policy is very small in this case. Batching all bays leads to an increase in pick time compared to the current situation if the capacity is less than six boxes.

For Algorithm 2 batching all bays works best if the batch capacity is more or equal to 9 boxes. If the capacity is small (4 or less), batching all bays leads to a decrease in savings compared to the current situation. Batching the last two bays or the last three bays results in a better performance for these kind of small batch capacities.

The results of Algorithm 3 are comparable to those of Algorithm 2. Only batching all bays works better than only the last three bays for a slightly smaller batch capacity (8 boxes) as in Algorithm 2 (9 boxes).

The final algorithm, Algorithm 4, works very well for small batch capacities. It is the only algorithm that leads to a saving in travel time for all the bay batching options for a small batch capacity. For a batch capacity of 4 or more it is best to batch all the bays instead of only a part of the bays. The reason for this is that the batches in Algorithm 4 are already limited to the same aisle. Therefore, a high pick density is ensured even when the batch capacity is low. However, if the batch capacity becomes larger than 9 boxes, the performance of this algorithm hardly increases. This is also caused by the limitation of creating batches only in the same aisle. There is only a limited number of boxes with picks in the same aisle, so at one point a higher batch capacity will not lead to more savings.

The increase in pick time compared to the current situation for small batch capacities in Algorithm 1, 2 and 3 can be explained by the fact that the savings in walking time do not outweigh the extra set-up time for making the batches. Especially for Algorithm 1, a low capacity will lead to a low pick density as picks can be located anywhere in the zone.

There are two batch capacities that are used to further evaluate the performance of the algorithms at the warehouse. These batch capacities are 9 and 12 boxes. The motivation for these capacities is as follows: Taking into consideration the width of the aisles it is determined that there can be three boxes next to each other on the pick cart. The pick cart can be three or four shelves high, which leads to the pick capacities of 9 (3 shelves x 3 boxes per shelf) and 12 (4 shelves x 3 boxes per shelf). These capacities were also confirmed by a company that sells order pick carts.

As the batch capacity will be at least 9 boxes, it is decided to only consider the ‘All bays’ option in the remainder of this research, i.e., to batch all the products regardless if they are slow or fast movers. The motivation behind this decision is that with a batch capacity of 9 boxes, batching all bays will be most favourable in all the algorithms, except Algorithm 1. The different outcome in Algorithm 1 will not lead to difficulties, as the performance of batching all bays and only the last three bays are already quite close to each other. Moreover, batching all bays (and therefore, all the products) is easier to implement for the company, as they do not have to make a distinction between boxes that have picks in the first bay and boxes that have picks in the last three bays.

5.2 Option 1: Single line boxes with current routing

After determining the batch capacities and bay batching option in Section 5.1, the performance of each algorithm can be evaluated more in depth to see which algorithm is most favourable for which batch capacity. The performance is evaluated on percentage of time saved compared to the current way of picking, the single order picking policy.

Table 4 shows the seconds per pick for the current situation, which is 28,134 seconds. Furthermore, it shows for both a capacity of 9 and 12 boxes for each algorithm the seconds per pick, calculated by the simulation. Also, the mean savings in seconds for all the algorithms compared to the current situation is given and the corresponding 95% confidence interval.

For a batch capacity of 9 boxes, the algorithm that leads to the most saving in seconds per pick is Algorithm 4. Implementing this algorithm will save on average 1,20 seconds per pick. For batch capacity 12, Algorithm 1 leads to the highest savings, which is 1,48 seconds per pick. Figure 10 gives a visual overview of the savings per batch capacity. In literature, the FIFO algorithm (on which Algorithm 1 is based) is generally used as a benchmark to measure the performance of other batching algorithms. Therefore, it can be concluded that, with a capacity of 9 or 12 boxes, algorithms 2 and 3 perform poorly. Algorithm 4 does perform well when the batch capacity is 9 boxes.

Table 4: Savings in seconds per pick per algorithm

Algorithm	sec/pick	Mean saving (sec)	Lower bound	Upper bound
			95% confidence interval	
Current	28,134			
Capacity = 9				
1	27,039	1,095	0,771	1,419
2	27,105	1,029	0,680	1,379
3	27,080	1,055	0,699	1,412
4	26,934	1,200	0,860	1,541
Capacity = 12				
1	26,654	1,481	1,139	1,821
2	26,943	1,191	0,815	1,568
3	26,933	1,202	0,821	1,583
4	26,844	1,290	0,921	1,661

Furthermore, in Figure 10 it can be seen that the performance of the three seed algorithms (Algorithms 2, 3 and 4) are very close together for each batch capacity. The performances of Algorithm 2 and 3 are almost the same for both batch capacities, which can be explained by the fact that both algorithms form batches based on distance. Only Algorithm 2 looks at the distance within a row, where Algorithm 3 looks at the shortest walking time in seconds. The last observation that can be made from the figure is that the performance of Algorithm 1 more strongly increases when the capacity increases, where the performance of the other algorithms only increases slightly when the capacity increases to 12 boxes. These only increase slightly as they can only form batches from the orders within the specific timeframe. Therefore, the total pool from which batches can be created is smaller.

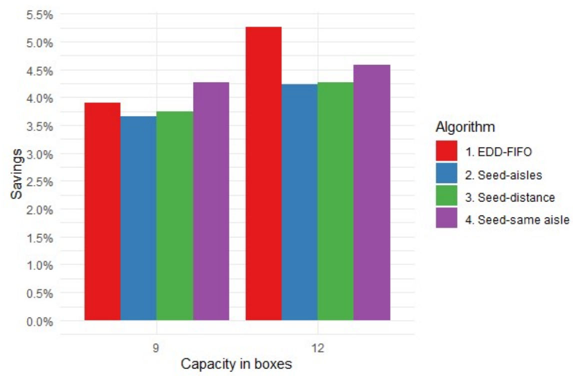


Figure 10: Savings in percentage per batch capacity and algorithm

5.3 Differences in day type

As mentioned in Section 2.4.1, days can be categorized in three groups based on the amount of pick lines per day. These three categories are ‘Busy’, ‘Normal’ and ‘Quiet’. A busy day has on average 7.093,5 pick lines. a normal day 6.115,5 and a quiet day 4.963,5. Of the 12 simulated days, there were 5 ‘Quiet’, 5 ‘Normal’ and 2 ‘Busy’ days. Looking at the average saving per day type in Table 5 it seems like ‘Normal’ days have a larger average saving when orders are batched according to Algorithm 1.

Table 5: Average savings per day type for Algorithm 1

Capacity	Day type			
	Quiet	Normal	Busy	Average
9	3,32%	4,77%	3,02%	3,87%
12	4,79%	6,11%	4,21%	5,24%

To see if there is a statistical difference per day type one-way analysis of variance (ANOVA) test can be used. This test assumes the that the observations are obtained independently and randomly from the population defined by the factor levels. Furthermore, as requirement the data of each factor level must be normally distributed, and these normal populations need to have a common variance. The data is tested for normality and common variance by the Shapiro-Wilks and Levine’s test (which can be seen in Appendix D).

For the Shapiro-Wilks test the null hypothesis could not be rejected. This does not mean that the data has a normal distribution, only that it could be normal. Looking at other indicators like the coefficient of variation (Table 6), it is questionable if the normality assumption required for the ANOVA holds. If the data is not normally distributed, a non-parametric test like the Kruskal-

Wallis test could be used. However, as an ANOVA is robust to violations of normality and the sample sizes per group are very small (5 ‘Quiet days’, 5 ‘Normal days’ and 2 ‘Busy days), an ANOVA test is used here to check for a statistical difference per day type.

Table 6: Standard deviation (SD) and coefficient of variation (CV) per day type for Algorithm 1

Capacity		Day type			
		Quiet	Normal	Busy	Average
9	SD	0,0133	0,0258	0,0143	0,0276
	CV	40%	54%	47%	71%
12	SD	0,0143	0,0267	0,0189	0,0279
	CV	30%	44%	45%	53%

An ANOVA is used to check for a significant variance between the means of three or more groups. This test has the following test hypotheses:

$$H_0: Means\ of\ the\ different\ groups\ are\ the\ same$$
$$H_1: At\ least\ one\ sample\ mean\ is\ not\ equal\ to\ the\ others$$

The results of the ANOVA test, at a 95% confidence level, per batch capacity can be seen in Table 7. For each batch capacity the p-value is larger than the significance value of 0,05. Therefore, H_0 cannot be rejected. Even though the averages per category in Table 5, seem to be very different, there is not enough evidence to suggest they are statistically different, and no further distinction will be made between the day types in this research. It is possible that there is no statistical difference due to the high standard deviation and small sample size of each category. Furthermore, for small sample sizes, it is difficult to obtain any meaningful statistical test results.

Table 7: ANOVA test per capacity

	Degrees of freedom	F-value	p-value (α = 0,05)
Capacity = 9			
Day type	2	0,880	0,448
Residuals	9		
Capacity = 12			
Day type	2	0,783	0,486
Residuals	9		

A ‘Normal’ day outperforms a ‘Quiet’ day in percentage saved as there are more pick lines on a normal day and therefore more combinations can be made for batching. ‘Busy’ performs worse than ‘Normal’ as more orders have to be picked from locations that are not frequently visited. So, operators have to walk more often to locations further away which takes more time.

5.4 Option 2: Single line boxes with new routing policy

The savings in Section 5.2 are calculated with the current, inefficient routing policy discussed in Section 4.7. A new routing policy based on the S-Shape heuristic was proposed and in this section the savings of using this policy are compared to the savings of the current routing policy.

It can be seen in Figure 11 that the new routing yields slightly higher savings for each combination of batching algorithm and batch capacity. The results in terms of which algorithm is the best for which capacity is comparable to the current routing. For a capacity of 9 boxes the difference between the four algorithms is very small, with Algorithm 4 having the best performance. Algorithm 1 clearly performs best for a capacity of 12 boxes.

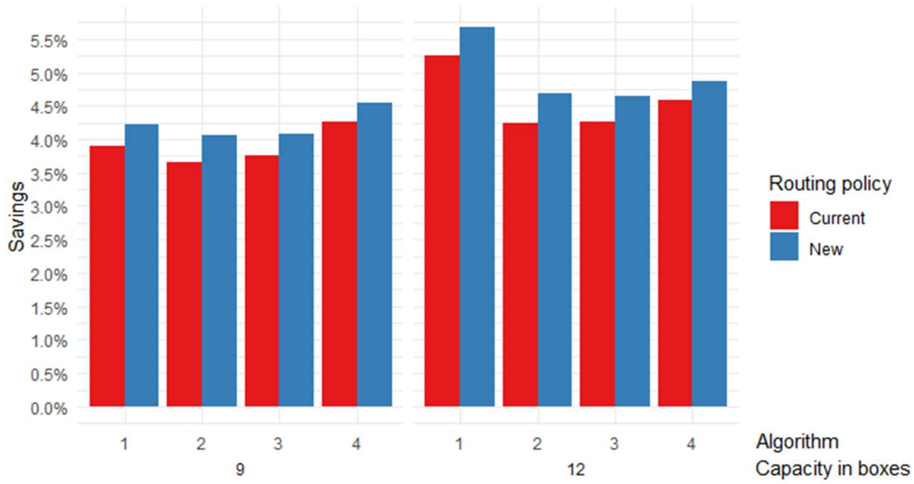


Figure 11: Percentage savings per pick from current routing policy compared to the new routing policy

For Algorithm 1 the difference in performance between the current and new routing algorithm is slightly larger than for the other Algorithms. This can be explained by the fact that Algorithm 1 creates batches randomly, where the other algorithms already consider the location of the products. When batches are created randomly, it is likely that an operator has to visit more aisles and will therefore benefit more from a logical routing policy.

The results of the new routing policy are compared to the results of the current routing to see if the new routing policy performs significantly better. For a batch capacity of 9, Algorithm 4 and Algorithm 1 are considered as Algorithm 4 had the highest performance for this batch capacity and Algorithm 1 is considered as its performance is quite close to Algorithm 4 but easier to implement. For the other batch capacity only the highest performing algorithm, Algorithm 1 is considered. From the results in Table 8 it can be seen that all the confidence intervals of the new routing policy are positive. This means that the new routing policy performs significantly better than the current one for the batching algorithms. However, this improvement in performance is quite small and it is therefore interesting to investigate if these savings outweigh the cost of implementing the new routing policy. This will be further analysed in Chapter 8.

Table 8: Savings in seconds per pick compared to the current routing policy

Algorithm	sec/pick	Mean saving (sec)	Lower bound	Upper bound
			95% confidence interval	
Capacity = 9				
4 - current	26,934			
4 - new	26,855	0,081	0,068	0,093
1 - current	27,039			
1 - new	26,945	0,096	0,081	0,110
Capacity = 12				
1 - current	26,654			
1 - new	26,535	0,120	0,105	0,135

5.5 Option 3: All pick lines with current routing policy

In the previous sections only the single line boxes were batched. Currently, it is difficult to batch multi lines as well due to the additional steps it takes to remove the unfinished boxes from the cart (Section 4.7). With an adaptation to the system, it is possible to make this step easier and to make it possible to batch all the lines in the repack area. However, this adaptation will require an investment, so it is interesting to see if the extra savings from batching all the lines outweigh the extra investment costs. Only Algorithm 1 (for capacity 9 and 12) and Algorithm 4 (for batch capacity of 9 boxes) will be considered in this part, as they had the highest performance for batching single line boxes.

In Figure 12 the performance of batching all the lines is compared to the performance of bathing only the single lines. Especially when the batch capacity is larger, batching all the lines leads to a higher performance. As can be seen in Figure 12, the difference between single lines and all the lines for a batch capacity of 1 is around 1%, while the difference for a capacity of 12 is close to 2%. Batching all pick lines leads to more savings compared to the single lines option as more lines are batched.

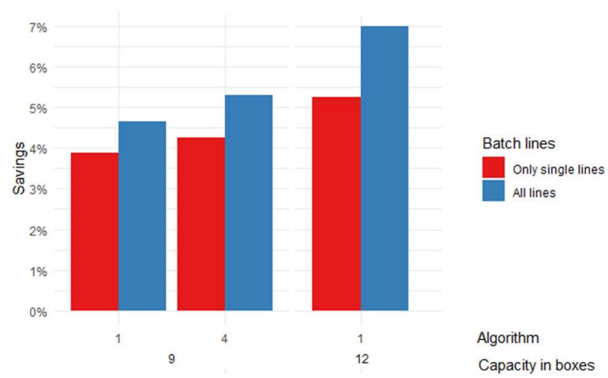


Figure 12: Percentage savings per pick only single lines compared to all lines

Table 9 shows the mean saving on seconds per pick and corresponding 95% confidence interval for batching all the lines compared to the current situation. Batching all lines does lead to a higher performance. However, in Chapter 8 it will be determined if this increase in performance outweighs the additional costs of the investment to make it possible to batch all the lines.

Table 9: Savings in seconds per pick for batching all boxes

Algorithm	sec/pick	Mean saving (sec)	Lower bound	Upper bound
			95% confidence interval	
Current	28,134			
Capacity = 9				
1	26,823	1,311	0,980	1,643
4	26,642	1,494	1,107	1,880
Capacity = 12				
1	26,163	1,971	1,614	2,328

5.6 Option 4: All pick lines with new routing policy

The last implementation option identified in Section 4.7 is to combine the new routing policy with batching all the pick lines. This option leads to an even further increase in savings seconds per pick, as can be seen in Table 10. Savings are increased as batching all pick lines will mean more boxes can be batched and each batch profits from the improved routing logic.

Table 10: All boxes in combination with the new routing policy

Algorithm	sec/pick	Mean savings (sec)	Lower bound	Upper bound	Mean saving (sec) (current routing)
			95% confidence interval		
Current	28,134				
Capacity = 9					
1	26,660	1,475	1,140	1,809	1,311
4	26,435	1,701	1,305	2,096	1,494
Capacity = 12					
1	25,947	2,187	1,826	2,549	1,971

5.7 Conclusion implementation options

In Figure 13 the savings in seconds per pick can be seen per option. For Algorithm 1, it shows the results for both capacities and, for Algorithm 4, the results for a capacity of 9 boxes.

When the batching capacity increases, the percentage saving increases as well. This makes sense as more pick lines can be combined into one pick route. For a batch capacity of 12 the difference in saving between batching all lines or only single lines becomes larger (almost 2% compared to a bit more than 1% for Algorithm 1). This indicates that when the company chooses carts with a higher batch capacity, it might be more beneficial to invest in enabling batching all the order lines. Moreover, the additional savings of changing the routing policy are higher for higher batch capacities, as more locations will have to be visited.

For a capacity of 9 the difference between the implementation options is larger for Algorithm 4 than Algorithm 1. The switching from the single lines – current routing to all lines – new routing saves almost an additional 2% in Algorithm 4 where it only saves close to 1,5% in

Algorithm 1. The savings for Algorithm 4 are higher as more pick lines can be batched and therefore more and better combinations can be formed. Algorithm 1 can only batch more when all the lines are batched, but it cannot create better combinations, as the batches are formed randomly.

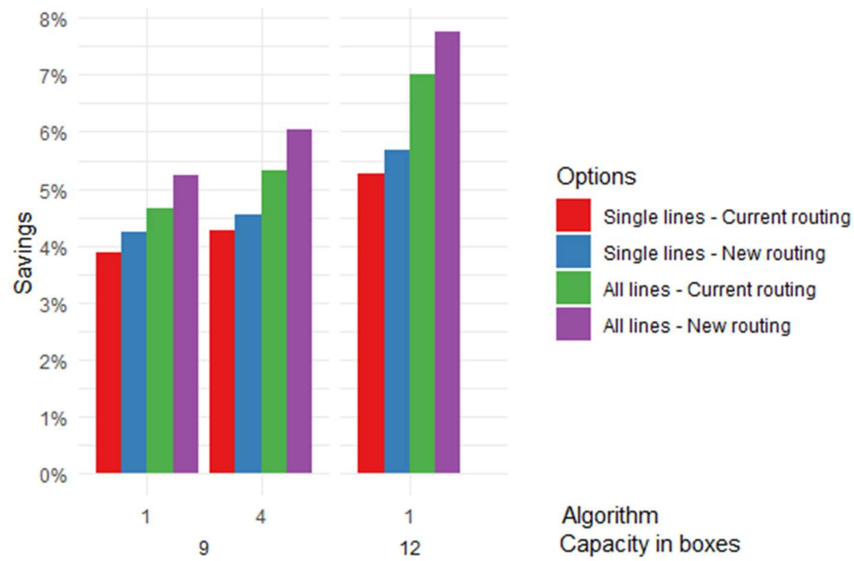


Figure 13: Savings per implementation option and batch capacity

5.8 Conclusion

When the pick capacity increases, it becomes less interesting to make a distinction to batch only the last bays of a warehouse. It is expected that the company will have pick carts with a capacity of 9 or 12 boxes. Only for Algorithm 1 it is more beneficial, with these capacities, to only batch orders from the last three bays instead of all the bays. However, as batching all the bays for a capacity of 9 and 12 still performs well, it is decided to focus only on batching all the bays. This decision is made because implementing batching for only the last three bays will require a lot of additional effort. In conclusion, no distinction is made in batching fast moving and slow-moving products.

Algorithm 1, based on EDD and FIFO, works best for larger batch capacities. For small capacities however, it works very poorly. If the capacity is very small (less than 6 boxes), batching even leads to an increase in average pick time, as could be seen in Figure 9 in Section 5.1. The increase in average pick time can be explained by the random formation of the batches. It is likely that with only a few boxes, the pick density is lower. Therefore, an operator will need to travel a long distance with only a few picks. The other algorithms work very well for small capacities, but their performance stagnates when the capacity increases. This stagnation can be explained by the division of the warehouse in zones and splitting the on-line batching problem into smaller off-line problems.

For the basic implementation option, where only single line boxes are batched and the routing policy is the current policy in place, Algorithm 4 has the highest performance for a capacity of 9 boxes. For a capacity of 12 Algorithm 1 has the highest performance. The performances of Algorithm 2 and 3 are quite close together, indicating that it might make no difference if one looks at distance in travel time between locations or at minimizing the number of aisles to be visited. This result is also in line with the research of Ho et al (2008), who found that there is no significant difference between their smallest number of additional picking aisles rule and their aisle distance-based rule. All algorithms for each capacity lead to a higher performance than the current situation, a single order picking policy.

There is no significant difference in the performance of each day type. Changing the walking policy to the S-Shape heuristic leads to the same best performing algorithms per batch capacity. Moreover, the performance with the new policy is slightly higher than the current routing policy. This difference is also significant. Changing the routing policy works better for an algorithm that randomly creates batches, like Algorithm 1. This can be explained by the fact that the pick locations will likely be further apart, and the operator will benefit more from a logical routing.

Batching all boxes leads to more savings than only batching single line boxes. This is because more boxes can be batched and therefore more and better combinations can be formed. However, it will be evaluated in Chapter 8 if this additional saving outweighs the investment needed to be able to batch all the lines. Especially since the additional saving of batching all the lines is quite small, 1 and 1.7% for Algorithm 1 and capacities 9 and 12 respectively.

As mentioned in Section 4.5 only three steps of the entire picking process are taken into account in this research. These steps were ‘Scan box’, ‘Walk to location’ and ‘Place box on conveyor’. Figure 14 shows the time distribution of the entire picking process. The first picture is the same as Figure 5 in Section 2.4.2, only with the three steps, ‘Scan box’, ‘Walk to location’ and ‘Place box on conveyor’, grouped into one step: ‘Travel time’. It can be seen that in the current process, the single order picking process, ‘Travel time’ accounts for 33% of the total time. The second picture shows the time distribution of the batch picking process, where the travel time is the result of Algorithm 1 for a capacity of 12 and Option 4. Here, the step ‘Travel time’ is only 31% of the total picking time. Hence, switching from a single order picking process to a batch picking process reduces the total picking time with 2%.

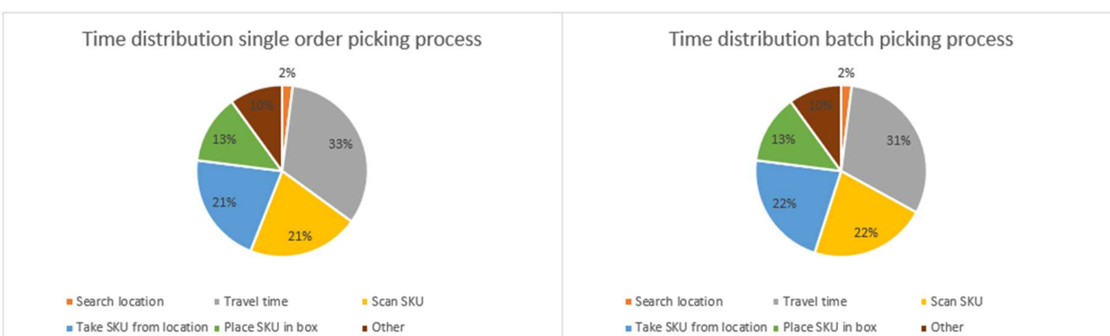


Figure 14: Time distribution of the single order and batch picking process

Now the fifth research question ‘*What is the expected improvement from the new methods?’* can be answered. For a batch capacity of 9 and 12 boxes the improvements from Algorithm 1 range from a saving of 3.89% to 5,26% respectively for the most basic option, Option 1. If all the lines are batched, this saving increases to 4.66% and 7.00%. Combining all the lines with the new routing policy the savings amount 5.24% and 7.77%. In monetary terms the most basic implementation option (Option 1) with the lowest batch capacity (9 boxes) will save the company approximately 5.250 euro a year. Implementing the most sophisticated option

(Option 4), and a capacity of 12 boxes, leads to a saving of approximately 10.500 euro on a yearly basis. In Chapter 8 an overview of the costs of implementing batching at the company is given.

6. Improve: Sensitivity analysis

To be sure that the models are robust for future changes, a sensitivity analysis is performed. With a sensitivity analysis it can be determined if the results of the algorithms hold for a change in one of the input parameters of the model and are therefore suited for future changes in the warehouse. In this section the following sensitivity analyses are performed: an increase in number of pick lines (Section 6.1), adjusting the timeframes (Section 6.2), a change in the travel times (Section 6.3), a change in the set-up time of batching (Section 6.4), and finally, a change in the batch capacity (Section 6.5). These analyses are chosen as they represent changes in most of the input parameters of Section 4.4. Changing these parameters will test the robustness of the algorithms, to see if they still hold if the input parameters change or if the algorithms can be applied to other warehouses. The analyses are conducted using the most basic implementation option, Option 1.

6.1 Change in pick lines

The performance of all basic scenarios described in Section 4.7 is evaluated for a change in the number of pick lines. The number of pick lines is increased and decreased by 10 and 20 percent. The average seconds per pick for each decrease and increase, and the corresponding 95% confidence interval can be seen in Table 11. This sensitivity analysis is relevant as in the future the number of pick lines in the warehouse is expected to increase. It is important to see if solutions that work now, also hold for the future. This sensitivity analysis also looks at the effect of a decrease in pick lines, so more general conclusions can be drawn from the results of this research.

Table 11: Average seconds per pick for a change in number of pick lines

	Mean seconds per pick	Lower bound	Upper bound
		95% confidence interval	
Regular	28,134	27,918	28,350
10% less pick lines	28,068	27,842	28,294
20% less pick lines	27,980	27,734	28,226
10% more pick lines	28,069	27,852	28,286
20% more pick lines	28,100	27,878	28,322

In Figure 15 the results of changing the number of pick lines for Algorithm 1 and 4 can be seen. Only these algorithms are shown in this sensitivity analysis as they were the best performing algorithms for a capacity of 9 and 12 boxes. As can be seen, an increase in pick lines leads to a small increase in the percentage saved on travel time, where a decrease in pick lines leads to a small decrease in savings. Especially for a capacity of 9 boxes this increase, and decrease is very small. For a capacity of 12 boxes, the increase in pick lines leads to a higher increase in savings. The higher saving for more pick lines can be explained by the fact that more pick lines provide more batch opportunities and carts will probably be filled more often to their full capacity. As the savings hardly change for a changing number of pick lines, it can be concluded that the developed algorithms are robust to a future change in pick lines.

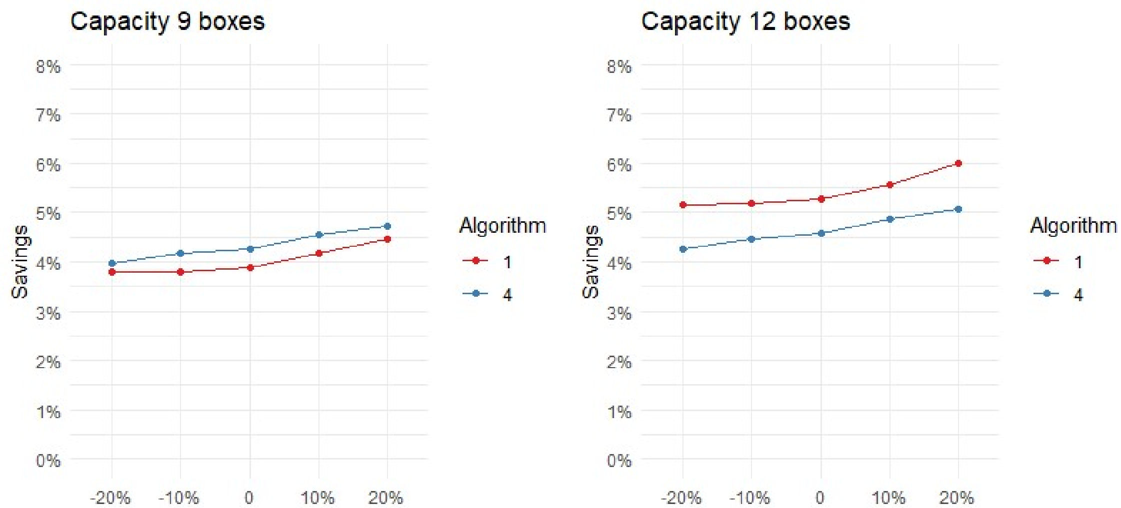


Figure 15: Change in number of pick lines

6.2 Adjusting timeframes

The four timeframes established in Section 4.3 limit the options to create batches, as only orders that are in the same timeframe can be batched. It would be interesting to see if larger timeframes can lead to a better performance of the algorithms and how the recommendations for the company change if they would be able to implement larger timeframes. Two timeframe options will be considered: three timeframes and no timeframes. Recall from Section 4.3 that there were four timeframes in the original analysis: all orders before 12:30h, orders between 12:30h and 14:00h, orders between 14:00h and 16:30h and all orders after 16:30h.

The three-timeframe option will merge the second and third group from the original analysis, resulting in the following timeframes: all orders before 12:30h, orders between 12:30h and 16:30h and all orders after 16:30h. These times are chosen as the 12:30h-14:00h timeframe is relatively small with respect to incoming orders.

Not having any timeframes changes the on-line batching problem into an off-line problem, where the batches can be formed after all orders are known. This would imply that the warehouse must change its way of working and not fulfil all the incoming orders on the same day, but on the next day. This is something that is not likely to happen in practice, however it is interesting to see how changing from an on-line to an off-line problem influences the results of the algorithms. These results can be interesting for other warehouses who are able to create the batches after all the orders are known, hence, to solve the off-line batching problem.

Figure 16 shows the results in savings in percentage for each timeframe. Comparing the performance of each timeframe option, the performance is higher if the timeframe restriction in relaxed. Switching from four to three timeframes does lead to some saving, although this saving is very small. Generally, the saving is slightly more than 4% per pick where the saving for four timeframes is a bit less than 4%. No timeframes, i.e., the off-line problem, does add to more additional saving. The saving can be as much as 7% per pick. Moreover, dividing the day into three timeframes only leads to a higher saving when the capacity is 9 boxes compared to the savings of Algorithm 1. For a capacity of 12, the savings from Algorithm 1 still outweigh the savings from Algorithm 2, 3 and 4 with three timeframes.

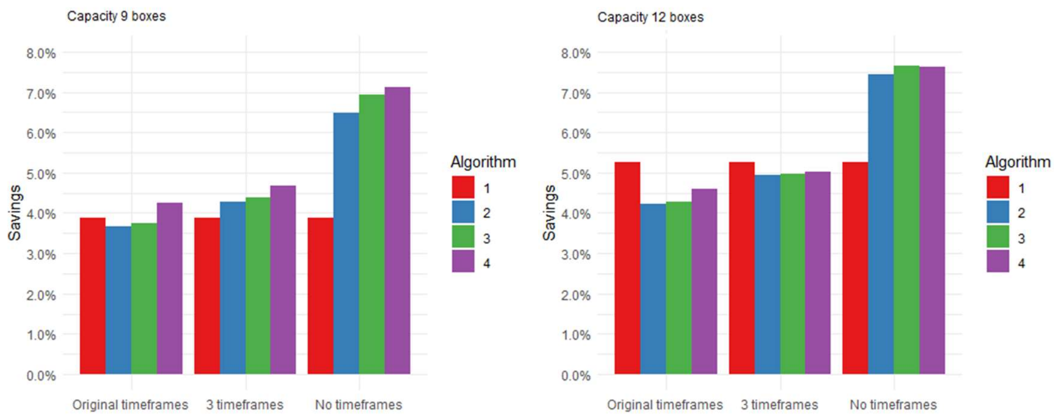


Figure 16: Savings per pick for each timeframe

When the batch capacity increases, the additional saving becomes less compared to the corresponding performance of Algorithm 1. This can be seen clearly in Figure 17. Algorithm 4 can increase the performance when the capacity is 9 boxes with an additional saving around 3.25% per pick when there are no timeframes, where it can only increase with 2.25% when the capacity is 12. Thus, having no timeframes will always lead to higher savings, no matter what batch capacity is used. However, where the potential additional saving compared to Algorithm 1 is very large for a capacity of 9 boxes (3.25%), it is smaller for a capacity of 12 boxes (2.25%). This makes sense intuitively, as a higher batch capacity means that more locations will have to be visited and these locations will be more dispersed in the zone.

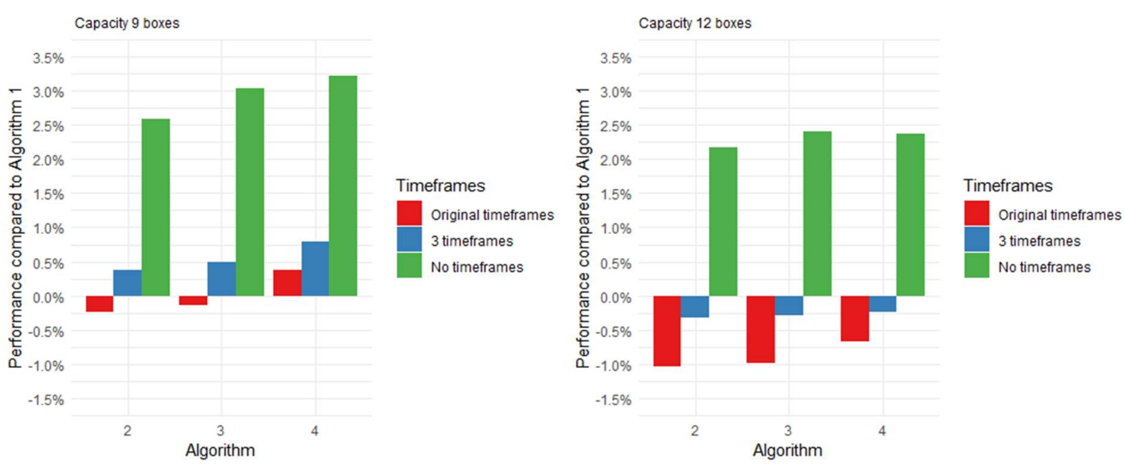


Figure 17: Additional savings per pick Algorithm 2, 3 and 4 compared to Algorithm 1

In conclusion, as it is unlikely that the company can start batching after all orders are known, Algorithm 1 is still most favourable for the batch capacity of 12. For a capacity of 9 Algorithm 4 is still the most favourable one. Having only three timeframes will increase the performance slightly compared to the initial four timeframes scenario. However, this increase will be close to only 0.2%. Therefore, it can be concluded that the results of the original analysis in Chapter

5 are quite robust when timeframes are implemented and that small changes in the timeframes do not lead to big changes in the results.

These results clearly show that algorithms that work well for the off-line batching problem cannot be copied and pasted into an on-line setting without proper research, as off-line solutions not necessarily work well in the on-line problem.

6.3 Changing travel time

An estimation of the time it takes for an operator to travel through the warehouse was made in Section 4.4. However, even though extensive measures were conducted to estimate these times, it is unsure if these travel times are correct and representative for every operator in the warehouse. Therefore, it is interesting to see how a change in travel time impacts the results on pick performance. This can also be interesting for warehouses who use robots and thus have a different travel time. In this sensitivity analysis the original walking time is increased and decreased by 10 and 20 percent. The results for a capacity of 9 and 12 boxes can be seen in Figure 18. The sensitivity analysis is only simulated for Algorithm 1 and Algorithm 4 as they were the best performing algorithms for these batch capacities.

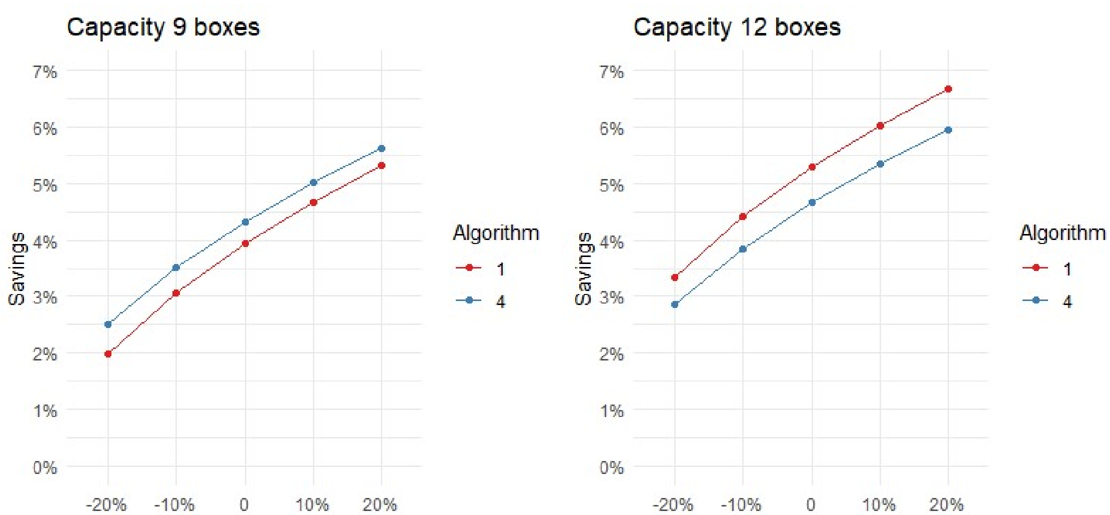


Figure 18: Change in travel time

If travel times increase, batching will increase the pick performance for both capacities. When the travel time increases, the single order picking policy will require a lot of travel time, so batching becomes more efficient. Moreover, the set-up times incurred when batching will be smaller compared to the travel time. A decrease in travel time leads to a smaller improvement in pick performance but it will still improve the performance compared to the current situation. The improvement will be less as a single order picking policy will not take as much time and the set-up times for the batches will more often outweigh the time saved in travel time.

For both capacities the preferred algorithm does not change when the travel time changes. The performance of Algorithm 4 has a slightly more horizontal line than Algorithm 1. This implies that the results Algorithm 4 are influenced by a lesser extend with a change in travel time than the results of Algorithm 1. An explanation for this difference could be that, in general, Algorithm 1 has longer pick routes as it can visit all locations in a zone in a single route. Algorithm 4 will always only visit one aisle. Therefore, a change in travel time will impact Algorithm 1 more.

Assuming using robots corresponds to a decrease in travel time, the percentage savings will be lower. Hence, warehouses that use robots in combination with the single order picking policy will benefit less from switching to a batch picking policy.

6.4 Changing set up times

As described in Section 4.4, the set-up times of creating a batch consists of a fixed time per batch and a variable time per box. These times are increased or decreased by 10 and 20 percent to see how these times influence the pick performance. It is interesting to see if improving the batch set-up process can lead to additional savings and what the implications are for warehouses with a different batch process and therefore a different set-up time. For this sensitivity analysis also only Algorithm 1 and 4 are simulated. The results can be seen in Figure 19.

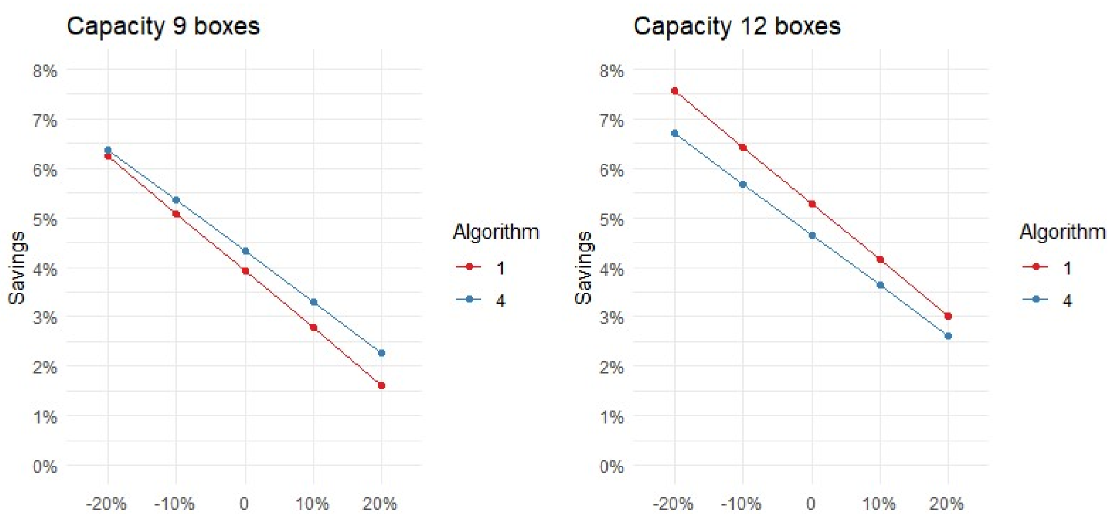


Figure 19: Change in set-up time

Set-up time has a strong influence on pick performance. A decrease in set-up time leads to an increase in pick performance. Set-up times are only incurred when batches are formed, therefore, when the set-up times decreases, the set-up times will have a smaller contribution to the total pick time and the performance will go up. The reverse is true for an increase in set-up time.

For both capacities the most favourable algorithm with the regular set-up times remains the most favourable when the set-up time changes. Algorithm 1 reacts more strongly to a change in set-up time than Algorithm 4. A reason for this can be the limitations of Algorithm 4 to form batches. Especially in the downstairs area where there are larger products, there is only a small demand of C1 and C2 boxes (the boxes considered for batching). As the formation of the batches is further restricted by the timeframes and by the same aisle rule, there are quite some aisles that only have one pick line per aisle per timeframe. This means that relatively more orders under Algorithm 4 are picked as a single order as under Algorithm 1. The set-up times are only incurred when there are batches, so two or more pick lines. Therefore, reducing or increasing the set-up time influences Algorithm 1 to a higher extent.

This sensitivity analysis shows that making the batch set-up process for an operator as easy and as fast as possible impacts the pick performance in a positive manner. It is therefore vital that warehouses keep reviewing their batch set-up process to make sure it is as efficiently as possible.

6.5 Different batch capacities

In Chapter 5 the results of the simulations were only presented for the most feasible batch capacities in the repack area, which are 9 and 12 boxes. These are realistic and feasible capacities for the warehouse in the case study. However, for other warehouses with different lay-out characteristics it is interesting to see how a different capacity influences the results. In this sensitivity analysis capacities between 3 and 24 boxes (with step sizes of 3 boxes) are simulated to see how changing capacities influences the performance in pick time for Algorithm 1 and 4. The results of this simulation can be seen in Figure 20.

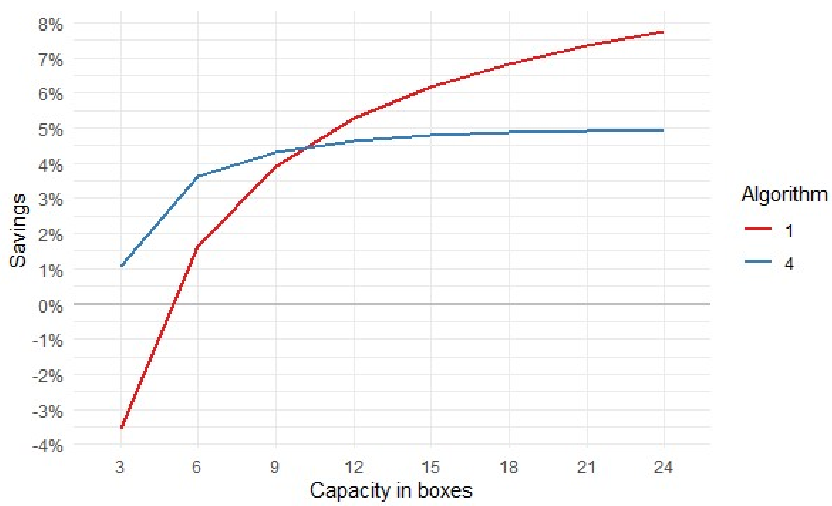


Figure 20: Savings per capacity

This figure clearly shows that for small capacities Algorithm 1 performs worse than Algorithm 4. Only after the capacity is higher than 9 boxes, Algorithm 1 performs better. Moreover, Algorithm 1 is more sensitive to a change in capacity. The random formation of the batches can turn out to be very inefficient for a low capacity, however when the capacity increases it becomes more likely that pick lines are located close together and that more efficient pick routes can be formed.

Algorithm 4 hardly improves when the capacity becomes larger than 9 boxes. This can be explained by the many limitations Algorithm 4 has in creating batches. Due to the division of orders into timeframes (Section 4.3) the available orders to batch are limited. The extra restriction of Algorithm 4 that all pick lines must be in the same aisle reduces the batching possibilities even further. However, for small capacities this algorithm works very well. Pick lines will always be in the same aisle, so the additional travel time from one line of the batch to another line is always shorter than picking them separately as the travel time to and from the aisle only has to be walked once.

6.6 Conclusion sensitivity analysis

There are several general conclusions that can be drawn from this sensitivity analysis. First, if the travel times are longer, batching orders will lead to more saving in travel time. Second, less set-up time for a batch will lead to more savings in travel time. And third, the higher the batch capacity, the less relevant sophisticated batching algorithms become. High capacities will reduce the pick densities of the batches of more ‘sophisticated’ algorithms, like Algorithm 2 and 3. Therefore, at a certain point, these batches will need to visit the same number of aisles as the FIFO algorithm and their performance converges.

The designed algorithms are not very sensitive to a change in pick lines, changes in the travel or set-up times, or switching from four timeframes to three timeframes. Even though the parameters change, the results in terms of a positive performance and which algorithm performs best for which capacity are quite similar. However, it should be noted that when travel or set-up times are higher or lower than assumed in this research, it could impact the potential amount of saving on the improvement in pick time.

The algorithms are very sensitive to changes in capacity (especially when capacity is smaller than 9 boxes) and the case when there are no timeframes, and the problem can be solved as an off-line problem. However, these two cases are very unlikely to happen. Firstly, as higher batch capacities always lead to higher savings, it is likely that the company prefers the highest batch capacity possible in their warehouse. As mentioned in Section 5.1 some tests were done to get an idea of the possible cart capacities at the warehouse and it was estimated that this could be at least 9 boxes. Second, as one of the drivers of e-business is to deliver as fast as possible it is very unlikely that they can change contracts with all their customers to deliver them not the next day, but two days after they place an order. Therefore, the results of the analyses are considered to be robust for situations where the capacity of a cart is at least 9 boxes, and the on-line batching problem has to be solved.

7. Improve: Scenario analysis

A scenario analysis is conducted to analyse the performance of the models in case the characteristics of the layout of the warehouse change. Three scenarios will be analysed in this chapter. First, a change in the ratio of the current ABC logic as discussed in Section 7.1. Second, removing all the last storage bays from the aisles in Section 7.2 and lastly, in Section 7.3, adding a storage bay to every aisle. These scenarios are chosen as they represent likely changes in the warehouse of the case study, and it is likely that other warehouses will differ in layout on these settings. The scenarios will be analysed only for the best performing algorithms in Chapter 5, which are Algorithm 1 and 4. Furthermore, these scenarios are only tested for the most basic implementation option, Option 1, where only the single line boxes are batched and pick routes are created by the current routing policy.

7.1 Scenario 1: Change in ABC storage

Currently, the idea is that products are stored in the warehouse according to the ABC storage policy. However, due to the volatile demand of the products and high costs of moving products in the storage area, this is not always the case. Table 12 shows the percentage of pick lines that is daily picked from each bay. A new division is proposed and simulated for Algorithm 1 and 4 to see how a change in ABC storage influences the performance of the algorithms. The results of the simulations can be seen in Figure 21. Note that this new ABC division is not thoroughly examined as this is not part of the scope of this research project. It is possible that this proposed division is not feasible in terms of available storage locations in each bay.

Table 12: Current and proposed pick line division per bay

	3 bay aisle		4 bay aisle	
	Current	Proposed	Current	Proposed
Bay 1	33%	40%	35%	40%
Bay 2	40%	40%	23%	30%
Bay 3	27%	20%	26%	20%
Bay 4	-	-	16%	10%

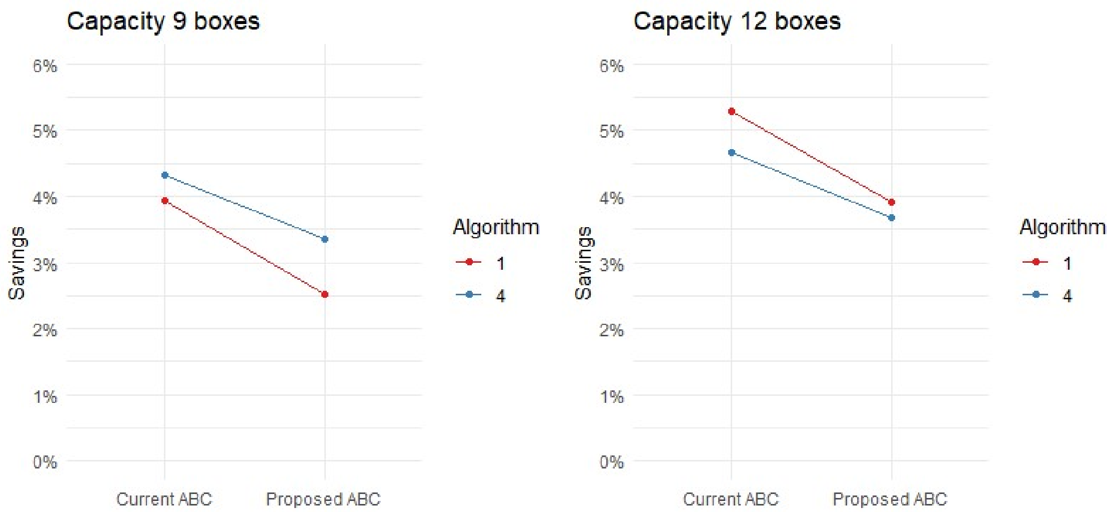


Figure 21: Difference current and proposed ABC

From these figures it can be seen that the improvement in pick performance is smaller for the proposed ABC than the current situation. However, it should also be noted that the average pick time for single order picking in the proposed ABC is 27,045 seconds where it is 28,134 seconds in the current situation. So, changing to a 'better' ABC already leads to a general improvement in pick performance.

For both capacities the best algorithm does not change when the ABC policy changes. For Algorithm 4 the proposed ABC leads to a higher performance compared to Algorithm 1. This can be explained by fact that Algorithm 4 only needs to visit one aisle. Operators will need to walk less often to the back of the aisle to retrieve a product. Algorithm 1 creates batches randomly, so a lot of aisles will be visited. As most of their picks are located closer in the aisle they will always have to enter and leave from same cross aisle and cannot benefit that much of making an S-shape through the aisle.

7.2 Scenario 2: Remove one storage bay

The second scenario is removing the last storage bay from each aisle. Removing the last bay will result in overall less walking time as operators will have to walk less far on average. However, it is interesting to see if the results of batching orders are similar to the current layout of the warehouse. This scenario is tested for the same amount of pick lines as the current scenario. As there is one bay less the division of pick lines over the bays had to be changed slightly. Aisles that went from three bays to two bays will have a division of 50% pick lines per bay in this scenario. Aisles that went from four to three bays get the same pick line division as the current three-bayed aisles, which is 33%-40%-27% (see Table 12 in Section 7.1). The results of this simulation can be seen in Figure 21.

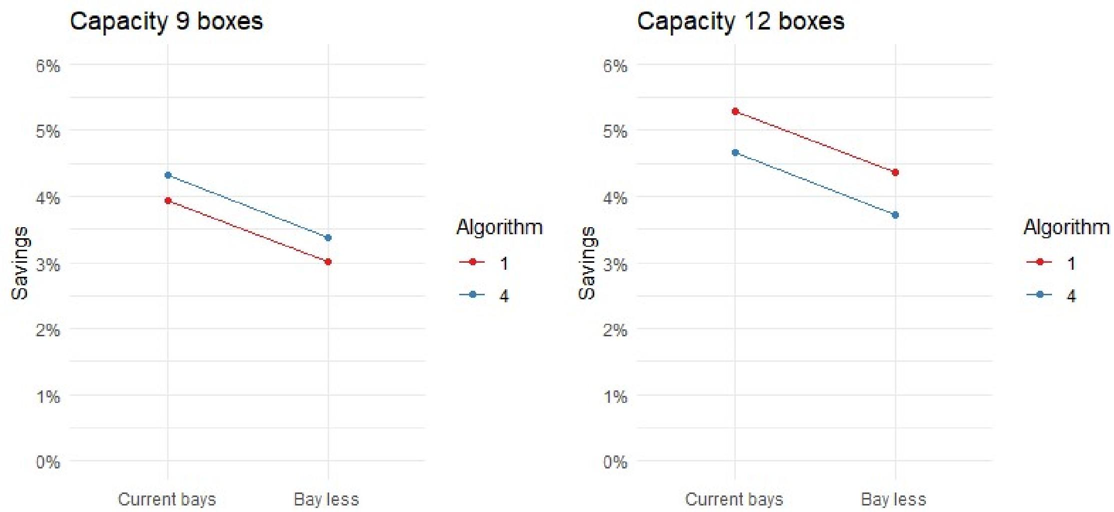


Figure 22: Difference current bays and one bay less

The figures show quite clearly that removing one bay from the warehouse leads to a decrease in potential saving in pick performance from batching. For both capacities the most favourable algorithm in the current layout remains the most favourable in the new layout scenario. Moreover, there seems to be no difference between algorithms in terms of their decrease in performance with one bay less compared to the current situation. The reason for the decrease in performance with the one bay less scenario could be that removing a bay already decreases the average pick time. Recall that the average pick time in the current layout for a single order picking policy is 28,134 seconds, removing the last bay decreases this time to 25,653 seconds. Furthermore, as the distances between products becomes smaller in this scenario, the extra savings from batching orders will be smaller.

7.3 Scenario 3: Adding one storage bay

The last scenario adds an extra storage bay to each aisle. If the company expects to extend their assortment in the future and needs more storage space, it is interesting to see if batching will still work well for them.

The simulation is tested for the same amount of pick lines as in the current scenario. As there is one bay more in each aisle the division of pick lines over the bays had to be changed slightly. Aisles that increased from three to four bays will get the same division of the current four-bayed aisles, which is 35%-23%-26%-16% (see Table 12 in Section 7.1). Aisles that increase from four to five bays will get the division 35%-23%-26%-8%-8%. The results of this scenario can be seen in Figure 22.

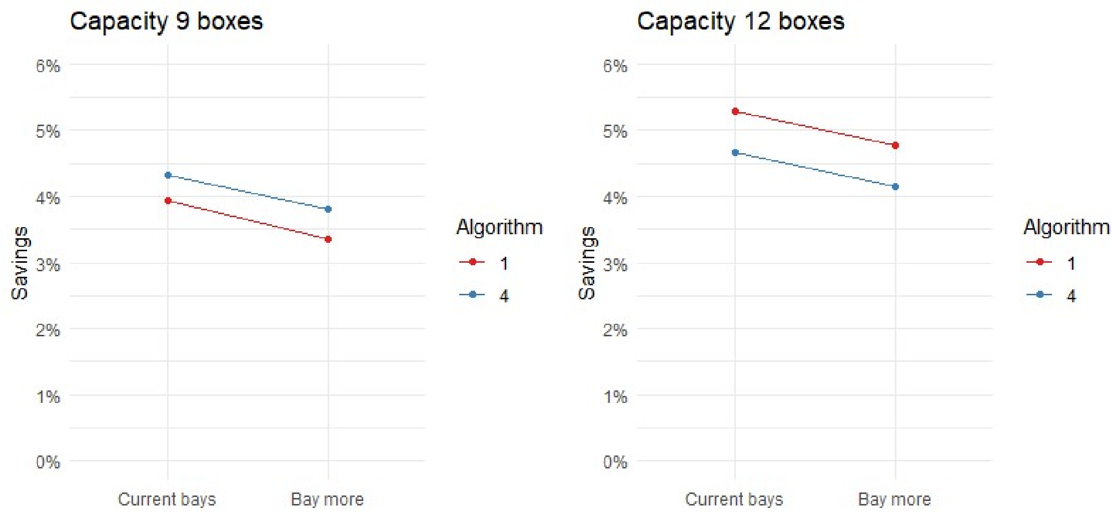


Figure 23: Difference current bays and one bay more

For this scenario similar conclusions can be drawn as the one aisle less scenario. For both capacities the most favourable algorithm in the current layout remains most favourable in the layout with one bay more per aisle. Also, with this scenario there is hardly any difference in the rate of decrease in performance between Algorithm 1 and Algorithm 4. Batching orders still improves the pick performance in the warehouse, although to a lesser extent compared to the current layout. The reason for this could be that adding an extra aisle leads to an increase in average pick time for a single order picking policy. This is 28,232 seconds for this scenario.

Another reason could be that adding an extra bay might increase the travel time of operators. Even though, generally slow-moving products are placed on the bays furthest from the conveyer and operators will not need to visit these locations often, they do have to walk around these bays to get to the next aisle when picking a batch. This will lead to a lot of additional travel time.

7.4 Conclusion scenario analysis

In conclusion, the different scenarios lead to similar results in terms of which algorithm is the best performing one. Even though the performance of the different scenarios tested is less than the performance of the current layout of the warehouse, batching is still able to improve pick performance compared to a single order picking policy for each scenario. Therefore, it can be concluded that the algorithms are robust for changes in the characteristics of the warehouse layout.

If there is a strong ABC policy in a warehouse, batching can be less beneficial as most of the products are located close to the I/O point. The set-up times to create a batch will be higher than to walk for each pick separately. Moreover, when the warehouse has many bays (so long aisles) with an ABC policy the performance of batching will be smaller. Operators will have to traverse the entire aisle before they can enter the next one. This will significantly increase the travel time. Removing one bay from the aisle can decrease the saving of batching as all the products will be located closer to the I/O point. This means that the set-up times to create batches might outweigh the savings in travel time of a batch.

The results in this section also show that the layout of the warehouse and the storage location assignment do influence the performance of batching algorithms. It is therefore important that warehouses keep this in mind when implementing new picking policies, to make sure that their layout settings match the picking policy to maximize their savings.

8. Improve: Required investments

To be able to answer the final research question: '*What are the expected required investments for the new methods?'* an overview of the costs of implementing batching at the repack area of the warehouse is given in this chapter. First the fixed costs are described, the costs that do not change irrespective of the chosen method and scenario (Section 8.1). Second, the specific costs per implementation option are estimated (Section 8.2). Finally, in Section 8.3, the costs are compared to the savings calculated in Chapter 5 and conclusions are drawn which option will be most favourable to the company in terms of return on investment. Finally, non-monetary investments are discussed in Section 8.4. As the results of Chapter 5 indicated that Algorithm 1 was most favourable for a capacity of 12 and the results of all algorithms were very close with a capacity of 9, only the costs associated with this algorithm are considered in this chapter. Moreover, only this algorithm is considered as it can be implemented without any additional costs and changes to the process.

8.1 Fixed investment costs

Fixed investment costs are the costs that are incurred irrespective of the batching method, implementation option and batch capacity. These costs are the costs of changing the current WMS to the new WMS that supports batching and small change requests that have to be made to make sure the new WMS works properly. Each of these costs is briefly discussed and an estimate of these costs is given. Besides the estimate, also an indication is given about the certainty of the costs. Note that all the costs are one-time investments. There are no yearly reoccurring costs associated with implementing batching.

Changing to new WMS

To change to the new WMS and to make sure there is enough server capacity to run the system there is a one-time investment needed of 22.500 euro. This amount is determined by the headquarters of the company and it is therefore likely that these costs will not change.

Small change requests

Currently, the screen configuration of the new WMS does not fit nicely in the scanners at the warehouse. The configurations have to be adapted for operators to work properly with the system. Moreover, some other small changes have to be made for the program to run smoothly on the RF scanners. These small changes to the program are expected to cost around 6.675 euro. This estimate is given by the system expert of the new WMS, so it is likely that the actual costs will be close to this estimate.

Total fixed investment costs

The total fixed investment costs can be seen in Table 13.

Table 13: Total fixed investment costs

Investment description	Costs
Changing to new WMS	€22.500
Small change requests	€ 6.675
Total	*€29.175*

8.2 Implementation option dependent investment costs

There are also some costs that depend on the chosen implementation option or batch capacity. The costs of the cart depend on the size and features of the cart, which depend on the desired batch capacity. As explained in Section 4.7 batching all lines is not possible at the moment, changing this in the WMS will only affect the scenarios in where all the boxes are batched. Changing the routing will also only affect the scenarios that take into account the new routing in the warehouse.

Purchasing carts

To be able to batch the orders, order pick carts will have to be purchased. The company expects to need 10 carts. Several quotations were requested from various companies. A cart with a capacity of 9 boxes will cost around 444 euro and a capacity of 12 will be 519 euro. This makes the total costs of capacity 9 4.440 euro and of capacity 12 5.190 euro.

All lines option in WMS

It is estimated by the software application engineer of Company X-Y that changing the function, so orders with lines in different zones can be batched too, costs around 24.000 euro. These costs are only relevant if the warehouse choses to implement batching for all boxes, so for Option 3 and 4. If it chooses to only batch the single lines these costs will not be incurred.

As changing this function might be beneficial for more warehouse sites, the costs can be split over these sites. However, it will take some time to sort out which sites would be interested in this change and how the costs will be split between the different sites. Therefore, for this research it is assumed that the warehouse will have to pay the full 24.000 euro required for this change by itself.

Change routing policy

It is estimated by the supervisor of the repack area that changing all the numbers takes 72 hours. This leads to a cost estimate of 1.500 euros. These costs are only incurred if the company choses to implement the new routing policy in the repack area, so Option 2 and 4.

8.3 Estimated investment per option

To see if the savings of the extra options outweigh the investments, the payback period in months and the Net Present Value (NPV) for a 10-year period (without considering the interest rate) is calculated per batch capacity. A 10-year period is chosen as this is the general time period used by the company for similar investments. These can be seen in Table 14 together with the costs and savings per implementation option.

Table 14: Results per batch capacity. With the lowest payback period and highest NPV in grey per capacity

	Capacity	
	9	12
Option 1: Single line - current routing		
Costs	€ 33.615	€ 34.365
Savings (per year)	€ 5.228	€ 7.065
Payback period (months)	77	58
Net Present Value (NPV)	€ 18.660	€ 36.285
Option 2: Single line - new routing		
Costs	€ 35.115	€ 35.865
Savings (per year)	€ 5.682	€ 7.635
Payback period (months)	74	56
Net Present Value (NPV)	€ 21.705	€ 40.485
Option 3: All lines - current routing		
Costs	€ 57.615	€ 58.365
Savings (per year)	€ 6.257	€ 9.407
Payback period (months)	111	74
Net Present Value (NPV)	€ 4.950	€ 35.700
Option 4: All lines - new routing		
Costs	€ 59.115	€ 59.865
Savings (per year)	€ 7.037	€ 10.439
Payback period (months)	101	69
Net Present Value (NPV)	€ 11.250	€ 44.520

It can be seen that the payback period of a capacity of 12 boxes is always lower than for 9 boxes. This is as the extra investments for a cart of 12 are only 75 euros more than a cart of 9. These 75 euros are negligible compared to the extra savings of batching 12 instead of 9 boxes. Furthermore, the new routing option (Option 2 and Option 4) always has a lower payback period and higher NPV than the current routing option (Option 1 and Option 3). This means that the additional savings of the new routing policy outweigh the extra costs to change this, no matter which batch capacity.

The lowest payback period, 56 months, is for Option 2 and a batch capacity of 12 boxes. However, looking at the NPV, Option 4 becomes the most favourable one. Even though the payback period is 13 months longer, the higher savings per year lead to a higher NPV after 10 years. Therefore, assuming the company choses for a capacity of 12, Option 4 is preferred. This means the company will have to invest in the batching all lines functionality of the WMS and change the routing policy in the warehouse.

If the company chooses for a capacity of 9 boxes, the results change. In this case the additional savings from the all lines investment are too small to outweigh the investment costs. For this capacity, Option 2 is most favourable both in terms of lowest payback period and highest NPV.

8.4 Non-monetary investments

Before batching can be implemented in the warehouse, the method has to be explained to the operators and they have to be trained properly to work with these methods. This section will list all the non-monetary investments the company will have to do before the operators can start batching the orders in the warehouse.

Training team coordinator and shift leaders

The team coordinator and shift leaders of the repack area have to be trained first to work with the new WMS. In a small presentation the features can be explained and what has to be done in case an error occurs. After the small presentation, they can go into the repack area to test the WMS in practice and see how it works. Also, in this way they will get to know the system sufficiently well so they will be able to explain this to the operators and help them in case they run into problems. This training can be given by the supervisor operations, who is the system expert for the new and current WMS.

Training operators

After the team coordinator and shift leaders are trained, the operators can be instructed. This training can be done in the same way as for the team coordinator and shift leads. It is recommended to train the operators in smaller groups so they have someone who can guide them through the system. This training can be given by the supervisor operations, the team coordinator and the shift leaders.

Increased supervision when the new WMS is implemented

It is recommended that the team coordinator and shift leads are more visible and have more time to answer questions when the new WMS is implemented. There can be a lot of questions from operators during the first few days they work with the system. The increased supervision only has to last a few days. The new WMS and the batching program are not very complicated, so it is expected that most people will be sufficiently proficient after a few days working with this system.

Training for inbound operators

Inbound operators are responsible for stocking the storage racks in the warehouse. Currently, the products are not always stored very neatly, and boxes might hang over the shelves. When outbound operators have to walk with a pick cart through these aisles, the cart can hit these boxes and the products might fall on the floor and get damaged. Or the outbound operators have more trouble getting through the warehouse as the aisles are smaller with the overhanging products. Therefore, it is important to show the relevance of stocking the products neatly in the bins to the inbound operators. This can be done giving a small training of half an hour maximum.

General ‘clean warehouse’ training for all employees

Sloppy storage of products in the aisles might cause trouble for picking with a cart, also other misplaced products will impede this process. Sometimes empty pallets block exits of aisles. If an operator is walking without a cart, this does not give many problems as they can just step over or around the pallet. However, if they are walking with a cart it takes a lot of time and effort to remove the pallet so they can exit the aisle properly.

8.5 Conclusion Improve phase

In this section the final two sub-questions will be answered. Sub-question 5 stated: *"What is the expected improvement from the new methods?"* The expected improvement depends on the chosen implementation option and the batch capacity. If a higher batch capacity is chosen, the expected improvement will be higher than for a lower batch capacity. Similarly, a more sophisticated implementation option leads to a higher saving. For a capacity of 9 boxes the difference in saving between the most basic and most sophisticated implementation option is 1,35% (3,89% for option 1 and 5,24% for option 4). Where for a capacity of 12 boxes the difference in savings increases to 2,51% (5,26% option 1 and 7,77% option 4).

Sub-question 6: *"What are the expected required investments for the new methods?".* Just like sub-question 5, this answer depends on the implementation option that is chosen. Table 14 in Section 8.3 gave an overview of all the savings and costs per option and batch capacity. Furthermore, it also showed the payback time and NPV (for a 10-year period, without considering the interest rate). From this table it was concluded that for a capacity of 9 boxes the savings of the batch all boxes function does not outweigh the extra investments. The savings of the new routing policy did outweigh the investments. Therefore, Option 2 was considered the best option for this capacity both in terms of payback period and NPV. For a capacity of 12 boxes, Option 2 had the lowest payback period. However, taking into account the NPV, Option 4 became the best. If Option 4 is implemented with a capacity of 12 boxes, the required one-time investments would be 59.865 euro. The savings will be slightly less than 10.500 euro per year.

Besides the investment costs, the outbound staff needs to be instructed and trained properly to work with the new WMS. For the other staff clear rules and regulations have to be communicated to make sure the aisles and the area around the aisle are kept clean and accessible for pick carts.

9. Conclusion, limitations, and recommendations

In this chapter the research question is answered in Section 9.1. In Section 9.2 the limitations of the research are discussed. Recommendations to the company are made in Section 9.3 and finally, suggestions for further research are given in Section 9.4.

9.1 Conclusion

In this section the research question: "*How can the pick performance in a low-level, manual, picker-to-parts warehouse be improved?*" will be answered by briefly answering the sub-questions.

Currently, the orders are picked with a single order picking policy. Furthermore, the picking process can be described as a low-level, manual, picker-to-parts warehouse. In 2020 the average daily pick line productivity per hour was 88,5.

From the literature followed that there are four common types of heuristics used to solve the OBP. These are priority rules, seed heuristics, savings heuristics, and metaheuristics. From these four methods only the priority rules and seed heuristics were suitable for the repack area.

These two rules were used as input to develop four algorithms. The first algorithm, Algorithm 1, uses a combination of priority rules, namely the EDD and the FIFO rule. The other three algorithms are based on seed heuristics. Algorithm 2 makes batches based on the nearest aisles, Algorithm 3 on the closest distance and in Algorithm 4 only pick lines from the same aisle are allowed to be batched. The algorithms are simulated using historical production data of the company of 12 days in 2021. The algorithms are simulated for four implementation options. Option 1 batches only the single line boxes with the current routing policy. Option 2 the single line boxes with the new proposed routing policy. Option 3 batches all the boxes with the current routing, and Option 4 batches all boxes with the new routing policy.

The results of the simulations showed that the expected improvement depends on the chosen implementation option and the batch capacity. If a higher batch capacity is chosen, the expected improvement will be higher than for a lower batch capacity. Similarly, a more sophisticated implementation option leads to a higher saving. For a capacity of 12 boxes Algorithm 1 has the highest improvement. If Option 4 is implemented, the savings will be 7,77% compared to the current situation

In order to see if the savings of the implementation options outweigh the investment costs, the payback period and NPV of each option is calculated. For a batch capacity of 12 the payback period for Option 2 is the lowest. However, the NPV is higher for Option 4 and therefore, it is recommended to invest in being able to batch all boxes and change the routing policy. This will require a one-time investment of 59.865 euro and lead to an annual saving of almost 10.500 euro.

In conclusion, the performance of the repack area can be improved by switching from a single order picking policy to a batching policy. The amount of improvement depends on the chosen capacity and implementation option.

9.2 Limitations

Several limitations of this research must be noted. Firstly, the outbound operation at the repack area is researched in isolation. However, in the morning there are also inbound operators working who stock the bins. The working hours of the inbound and outbound operators slightly overlap and during this time picker blocking can occur. The inbound operators also use the same RF program and the same routing policy that is used for picking products, to store products in the warehouse. Therefore, changes in the logic at outbound can also influence the performance at inbound. More research is needed to see what the impact of these changes will be on the performance of the inbound department, as even the system experts are unsure about the impact of the changes. The impact can best be tested by changing the routing policy in one zone and see what the impact of this change is in this zone. Once this is understood properly, the routing logic can gradually be implemented in the other zones.

In this research, it is assumed that only the small boxes are batched. This means that the big boxes still have to be picked by the current single order picking policy. Batching only works with the new system but this system does not support a single order picking policy. This means that the large boxes still have to be picked with the current WMS. The company has to research how the operators can work with these two systems simultaneously. This can be done for example by giving each operator two RF scanners, one with the new and one with the current WMS. The operator can now pick the large boxes with the current system and when there are enough small boxes, place them on a cart and batch pick these with the new WMS. Or two separate teams within the outbound group can be made. Where one team only picks batches with a cart and the other team the large boxes with the current WMS. However, with this solution each zone can have multiple operators working at the same time, which increases the chances on picker blocking.

The data in this research consisted of 12 production days in February and March 2021. This period was chosen to represent the most current layout of the warehouse. However, this means that the peak season in December is not taken into account. To make sure the results of this research were generalisable for this period, the data was divided into three categories (‘Busy’, ‘Normal’ and ‘Quiet’ as mentioned in Section 4.4). The number of pick lines of the ‘Busy’ days match the average number of pick lines in December 2020. However, as the number of ‘Busy’ days in this period was only 2, it is hard to draw general conclusions for this period.

If the warehouse chooses to implement batching, they have make sure the aisles and cross aisles are kept clean and that the products are stocked products neatly. Currently, a lot of products hang over the sides of the shelves or there are empty pallets placed at the end of an aisle blocking the path for a cart.

Furthermore, this research only focussed on travel time. The VSM in Section 2.4.2 (Figure 5) showed that travel time only accounts for 28% of the entire picking process. Other time-consuming steps like ‘Take SKU from location’ or ‘Scan SKU’ can be investigated to see if the time of these steps can be reduced.

As mentioned in Section 2.4.3, it is important that the quality of the process does not decrease. When batch picking is implemented, several boxes will be picked simultaneously, and it is likely that more mistakes can be made. For example, operators placing the product in the wrong box. Once the chosen batch method is implemented, the company will have to keep a close eye on the error percentages at the incorrect weight lane and at the customer, to ensure the high quality of the process.

The final limitation is that the breakdown periods of the machines are not considered. If the machines breakdown, the boxes cannot be delivered at the right time at the right zone. In this case, the operators will form different and, most likely, sub-optimal batches compared to the ones suggested by the algorithms. This can lead to less savings than the savings calculated in this research. Or there are not enough boxes available at a zone due to the breakdown, and operators will create batches with less boxes than the batch capacity allows.

9.3 Recommendations

It is recommended to the company to switch from the current single picking policy to a batch pick policy, as this will improve the pick performance. A batch capacity of 9 or 12 boxes seems to be feasible with respect to the size of their aisles. As a higher capacity gives a higher performance, it is recommended to purchase carts that fit 12 boxes.

For a capacity of 12 boxes, Algorithm 1 has the best performance. Therefore, it is recommended to implement this Algorithm in the warehouse. Another advantage of this algorithm is that it can be implemented right away and does not require any additional investments.

Even though the additional savings of Option 4 outweigh the costs of these investments, it is recommended to start with the most basic option, Option 1. This recommendation is made as some more time is needed to investigate the implications of the additional investments and to implement these changes. Changing the routing policy will take some time and it is recommended to do this zone by zone. In this way potential mistakes in one zone can be spotted and corrected, and the mistake will not be made in each zone. To be able to batch all the pick lines, a change request has to be made at the software department of the company. Making this change will take some time and a substantial investment. The company can delay making this change request and first try to find other warehouse sites that are interested in the same change request. If the request is made by several warehouse sites, the costs of the change will be split across these sites. Therefore, delaying implementing Option 4 can result in less investment cost of this option and a lower payback period.

Furthermore, it is recommended to investigate thoroughly the implications of changing the routing policy on the putaway logic. The inbound and outbound department use the same system and routing logic. Changes in the logic for outbound sequence also change the sequence in which items are stocked by the inbound department. It is important to investigate how these changes influence the outbound sequence, as it might decrease the performance for this department.

Batching can also be used to stock products. In the current process, products are stocked in a similar way to which products are picked. Products arrive in the zone with the conveyor and an inbound operator takes the products from the conveyor and places them in the right storage bin. Also for this process, it might be possible that the products are placed on a cart and the operator stocks multiple products in one run. However, this is something that has to be investigated.

The last recommendation is to see how the time of the other steps in the picking process can be reduced. This research only focussed on the travel time. However, in Figure 5 (Section 2.4.2) it could be seen that the picking process consisted of sever other steps like 'Scan SKU', 'Take SKU' and 'Place SKU in box' which took 22%, 22% and 13% respectively. It would be interesting to see how these steps can be reduced and how much they can decrease the total picking time.

9.4 Future research

In this section, two areas for future research are identified. The first area is to research the effects when zoning is integrated with other picking policies. The second is solving the on-line batching problem. Both areas will be discussed briefly.

One of the research gaps identified in the literature review of Janssen (2020) was that the effects of zoning are hardly integrated with other methods. This area deserves more attention as zoning can have a serious effect on the performance of the other picking policies. When the total pick area is divided into zones, the batching problem is automatically divided into smaller problems, i.e., one batching problem per zone. As the area of a zone and therefore the number of pick lines becomes smaller, other conclusions might be drawn concerning the effect of picking policies than when the total area of the warehouse is considered. Moreover, the formation of batches can be affected by boxes that have pick lines divided over multiple zones.

The second area that is identified for future research is the on-line batching problem. As mentioned in Section 4.3, the on-line batching problem has gotten far less attention than the off-line problem, even though the on-line problem is more realistic in practice. The sensitivity analysis on timeframes in Section 6.2 showed that the algorithms 2, 3 and 4 performed a lot better compared to Algorithm 1 when there are no timeframes (the off-line problem). However, when there were timeframes (on-line problem), the performance was worse. This shows that well performing heuristics in the off-line problem are not always suited to solve the on-line problem. More research is needed in this area to investigate which algorithms are suited for on-line problems and under which conditions.

References

Alipour, M., Zare Mehrjedrdi, Y., & Mostafaeipour, A. (2020). A rule-based heuristic algorithm for on-line order batching and scheduling in an order picking warehouse with multiple pickers. *RAIRO - Operations Research*, *54*(1), 101–117. https://doi.org/10.1051/ro/2018069

Cergibozan, Ç., & Tasan, A. S. (2019). Order batching operations: an overview of classification, solution techniques, and future research. *Journal of Intelligent Manufacturing*, *30*(1), 335–349. https://doi.org/10.1007/s10845-016-1248-4

Cergibozan, Ç., & Tasan, A. S. (2020). Genetic algorithm based approaches to solve the order batching problem and a case study in a distribution center. *Journal of Intelligent Manufacturing*, *2015*. https://doi.org/10.1007/s10845-020-01653-3

Chen, F., Wei, Y., & Wang, H. (2018). A heuristic based batching and assigning method for online customer orders. *Flexible Services and Manufacturing Journal*, *30*(4), 640–685. https://doi.org/10.1007/s10696-017-9277-7

De Koster, M. B. M. (2012). Warehouse Assessment in a Single Tour. *Springer 4*, *9781447122746*, 457–473. https://doi.org/10.1007/978-1-4471-2274-6_17

De Koster, M. B. M., Poort, E. S. V. Der, & Wolters, M. (1999). Efficient orderbatching methods in warehouses. *International Journal of Production Research*, *37*(7), 1479–1504. https://doi.org/10.1080/002075499191094

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. https://doi.org/10.1016/j.ejor.2006.07.009

Gibson, D. R., & Sharp, G. P. (1992). Order batching procedures. *European Journal of Operational Research*, *58*(1), 57–67. https://doi.org/10.1016/0377-2217(92)90235-2

Ho, Y. C., & Tseng, Y. Y. (2006). A study on order-batching methods of order-picking in a distribution centre with two cross-aisles. *International Journal of Production Research*, *44*(17), 3391–3417. https://doi.org/10.1080/00207540600558015

Ho, Ying Chin, Su, T. S., & Shi, Z. Bin. (2008). Order-batching methods for an order-picking warehouse with two cross aisles. *Computers and Industrial Engineering*, *55*(2), 321–347. https://doi.org/10.1016/j.cie.2007.12.018

Hsu, C. M., Chen, K. Y., & Chen, M. C. (2005). Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Computers in Industry*, *56*(2), 169–178. https://doi.org/10.1016/j.compind.2004.06.001

Janssen, J. (2020). *1ML05 – Preparation Master Thesis 1: Literature Study.* Unpublished Manuscript, Eindhoven University of Technology. 1–27.

Kübler, P., Glock, C. H., & Bauernhansl, T. (2020). A new iterative method for solving the joint dynamic storage location assignment, order batching and picker routing problem in manual picker-to-parts warehouses. *Computers and Industrial Engineering*, *147*(August 2019), 106645. https://doi.org/10.1016/j.cie.2020.106645

Landry, M., Malouin, J. L., & Oral, M. (1983). Model validation in operations research.

European Journal of Operational Research, *14*(3), 207–220. https://doi.org/10.1016/0377-2217(83)90257-6

Lolkema, F. (2020). *Optimizing the order batching process in an e-fulfilment center* (Master thesis, Eindhoven University of Technology, Eindhoven, the Netherlands). Retrieved from: https://research.tue.nl/

Sargent, R. G. (2010). Verification and validation of simulation models. *Proceedings - Winter Simulation Conference*, *January 2011*, 166–183. https://doi.org/10.1109/WSC.2010.5679166

Scholz, A., & Wäscher, G. (2017). Order Batching and Picker Routing in manual order picking systems: the benefits of integrated routing. *Central European Journal of Operations Research*, *25*(2), 491–520. https://doi.org/10.1007/s10100-017-0467-x

Scholz, André, Schubert, D., & Wäscher, G. (2017). Order picking with multiple pickers and due dates – Simultaneous solution of Order Batching, Batch Assignment and Sequencing, and Picker Routing Problems. *European Journal of Operational Research*, *263*(2), 461–478. https://doi.org/10.1016/j.ejor.2017.04.038

Tsioptsias, N., Tako, A., & Robinson, S. (2016). Model validation and testing in simulation: A literature review. *OpenAccess Series in Informatics*, *50*(6), 6.1-6.11. https://doi.org/10.4230/OASIcs.SCOR.2016.6

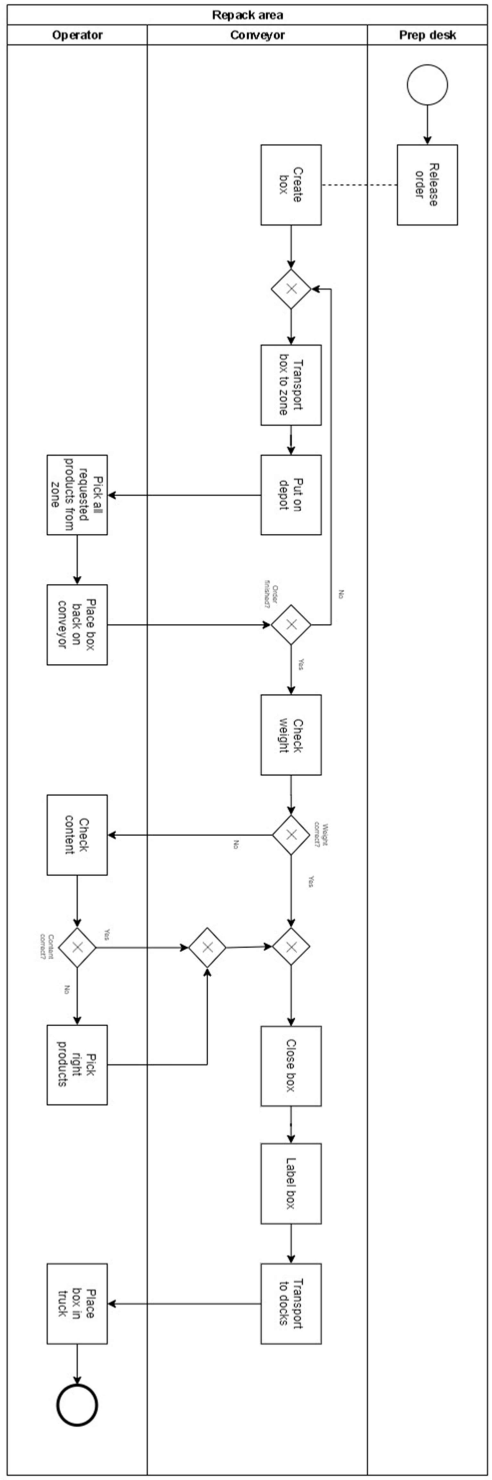
van Gils, T., Caris, A., Ramaekers, K., Braekers, K., & de Koster, R. B. M. (2019). Designing efficient order picking systems: The effect of real-life features on the relationship among planning problems. *Transportation Research Part E: Logistics and Transportation Review*, *125*(August 2018), 47–73. https://doi.org/10.1016/j.tre.2019.02.010

van Gils, T., Ramaekers, K., Caris, A., & de Koster, R. B. M. (2018). Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review. *European Journal of Operational Research*, *267*(1), 1–15. https://doi.org/10.1016/j.ejor.2017.09.002

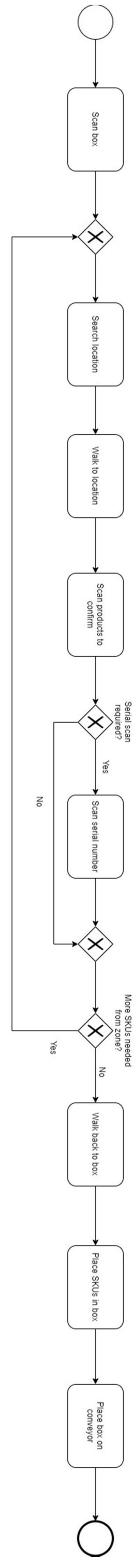
Zhang, J., Wang, X., Ruan, J., & Chan, F. T. S. (2017). On-line order batching and sequencing problem with multiple pickers: A hybrid rule-based algorithm. *Applied Mathematical Modelling*, *45*, 271–284. https://doi.org/10.1016/j.apm.2016.12.012

Appendix

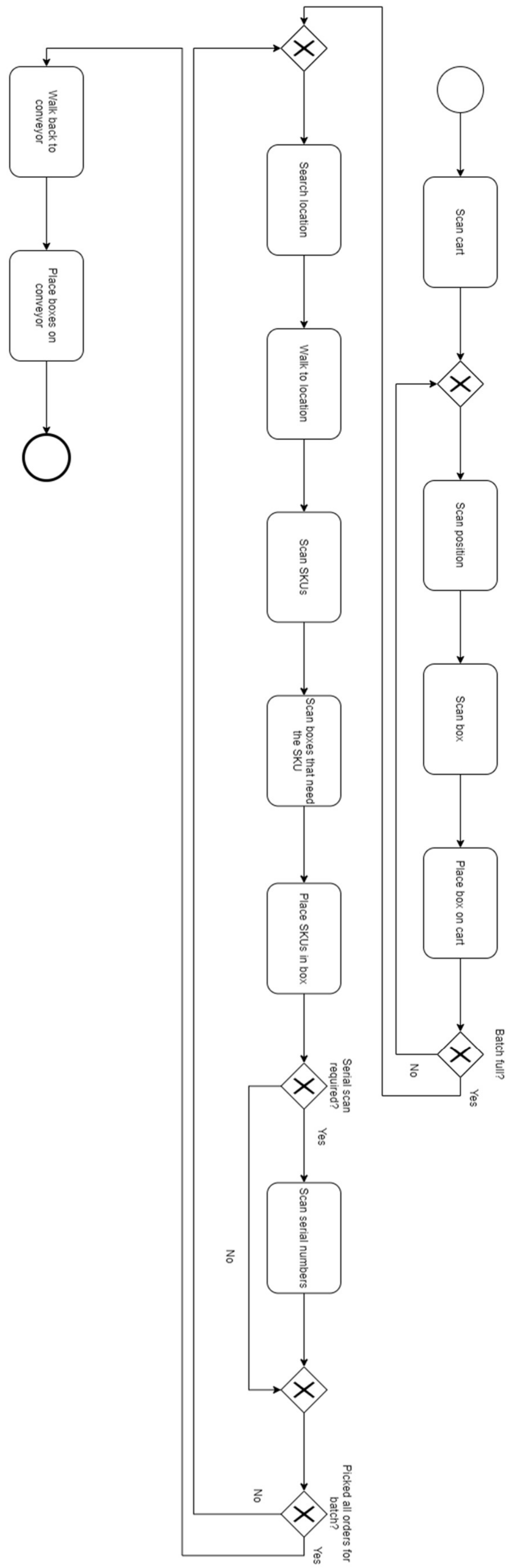
A. Conveyor process



B. Current pick process



C. Batch pick process



D. Tests for ANOVA assumptions

As mentioned in Section 5.3 the assumptions of an ANOVA are that the data is normally distributed and have a common variance. The normality assumption can be checked with the Shapiro-Wilk normality test. The hypotheses of a Shapiro-Wilk normality test are as follows:

$$H_0\text{: } The\ sample\ is\ normally\ distributed$$
$$H_1\text{: } The\ sample\ is\ not\ normally\ distributed$$

The results of this test for a 95% confidence level can be seen in Table 15. The p-values of each capacity level exceed 0,05. Therefore, H_0 cannot be rejected and there is no indication that the normality assumption is violated.

Table 15: Shapiro-Wilk normality test

Capacity	W	p-value (α = 0,05)
9	0,9898	0,9997
12	0,9869	0,9984

The second assumption, homogeneity of variances in the different groups, is checked with the Levene's Test. This test is most commonly used to assess the homogeneity of variance. The hypotheses of the test are as follows:

$$H_0\text{: } The\ variance\ is\ equal\ across\ all\ groups$$
$$H_1\text{: } The\ variance\ is\ not\ equal\ across\ all\ groups$$

Levene's Test assumes that the observations are independent and that the test variable is quantitative. These assumptions both hold as the observations are production days at the warehouse, which are independent, and the test variable is percentage saving. This is neither nominal nor ordinal. The results in Table 16 show the results for a 95% confidence level. As the p-values are higher than the significance level of 0,05, H_0 cannot be rejected there is homogeneity of variances in the different groups. Therefore, it can be concluded that the data meets the criteria of an ANOVA test.

Table 16: Levene's test for homogeneity of variance

	Degrees of freedom	F-value	p-value (α = 0,05)
Capacity = 9			
group	2	1,3585	0,3051
	9		
Capacity = 12			
group	2	0,9700	0,4154
	9		