

MASTER

Improving picking efficiency in a warehouse with multiple floors at Docdata NV

Verschure, A.P.C.

Award date:
2014

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Eindhoven, August 2014

**Improving picking efficiency in a warehouse
with multiple floors at Docdata NV.**

By
Anniek Verschure

BSc Industrial Engineering – TU/e 2013
Student identity number 0715072

in partial fulfilment of the requirements for the degree of

**Master of Science
in Operations Management and Logistics**

Company Supervisor:

S. Hooglugt, Docdata NV

H. Dankers MSc., Docdata NV

TU/e Supervisors:

dr. ir. R.A.C.M Broekmeulen, TU/e OPAC

ir. dr. S.D.P. Flapper, TU/e OPAC

TUE. School of Industrial Engineering.

Series Master Theses Operations Management and Logistics

Subject headings: Batching, Picking, Order Picking, E-commerce

Abstract

This master thesis describes the result of a research that designs a methodology to create and pick batches in a multiple floor warehouse and analyzes the effect of multiple floors on the pick performance. Simulation has shown that the methodology designed results in both increased pick and due date performance. Using this methodology, the effect of dividing a warehouse in multiple floors is evaluated. Sensitivity analysis shows that an increased number of floors can be beneficial for the pick performance when the floor switch time is low in comparison to the number of cross aisles. On the other hand, locating all pick locations on one floor is more beneficial when the floor switch time is high in comparison to the number of cross aisles.

Executive summary

This research is performed at Docdata fulfillment, a logistic service provider (LSP) which takes care of storage and handling of products sold online. The warehouse is composed of multiple floors located above each other connected by multiple elevators.

Problem description

The aim of this research is to reduce the time devoted to the labor-intensive operation: picking. The time devoted to picking is composed of four elements. These are the time to start up a batch, including receiving empty pick carts and printing batch forms, time to extract items from the shelves, travel time and floor switch time. The total time devoted to picking can be influenced by three hierarchical dependent processes: batch releasing, batch clustering and pick routing. The research assignment is defined as:

*Design a methodology to route, cluster and release batches
such that the time devoted to picking is minimized.*

Research design

The three processes have a strong hierarchical dependence, such that a decision made in one phase of the hierarchy strongly influences the decisions that have to be made in the other phases of the hierarchy. Since the decisions made on higher hierarchical levels reduce the design options in the lower levels, the design is made bottom up. Given the incoming orders and their associated locations, an appropriate pick routing strategy is the currently used S-shape strategy. According to this strategy, an order picker enters every aisle where an item has to be picked and travels the entire aisle. Given that batches are routed S-shape, the seed algorithm is preferred according to De Koster et al. (1999). A seed algorithm consists of two steps. First, the seed (or initial) order has to be selected from those orders not yet added to a route. Second, not yet selected orders have to be added to the seed order until the pick device capacity is filled. Given that orders are batched, the aim of the release policy is ensuring that all orders are handled in time. More efficient batches can be made when more information is available. A batch can only start when an order picker is available. A constant workforce is assumed and the batch creation time is negligible. Therefore, batches should only be created when an order picker is available. Because quantity based release policies outperform time based release policies, the release policy should be quantity based. In addition, it should also be assured that all orders are handled in time, and therefore a time based release policy is needed as well. After 11 PM, no additional orders will come in, and thus the time based cut-off value is 11 PM.

Additional to the methodology used at Docdata fulfillment, six scenarios are defined and evaluated on the pick and due date performance. The pick performance is defined as the total time needed per item and the due date performance is defined as the percentage of batches which finish after 12.30 AM. The scenarios deviate from each other due to different parameter values for seed selection and order addition respectively. The possible parameter values for both the seed selection and order addition are described below:

Seed selection methodologies:

- Docdata seed selection: The initial order is the order with the lowest alphabetic location number on the first order line.
- Longest processing time (LPT): The initial order is the order with the individual longest processing time (processing time consists of travel time, floor switch time and extraction time).
- Floor longest processing time (FLPT): The initial order is the order located at the minimum number of floors, with the longest individual processing time.

Order addition methodologies:

- Docdata order addition: Add the order with the lowest alphabetic location number on the first order line.
- Savings order addition: Add order j that maximizes the savings. (Savings are defined as the difference of retrieving the seed order and order j individually compared to retrieving the seed order and order j simultaneously).
- Floor savings order addition: Add the orders to the seed order which contain no more floors than the seed order, based on the maximum savings.
- Longest processing time (LPT): Add the orders to the seed which have the longest individual processing time.

Results

Simulation has shown that seed selection based on LPT outperforms Docdata seed selection in terms of due date and pick performance. In addition, seed selection based on LPT outperforms seed selection based on FLPT on pick performance; however, the opposite is true in terms of due date performance. Because seed selection based on LPT exceeds the due date performance norm, seed selection based on LPT is preferred.

Simulation has shown that the savings and floor savings order addition outperform the LPT and Docdata order addition both in terms of pick and due date performance. The floor savings order addition due date performance exceeds the savings order addition due date performance, but both are above the desired norm. Sensitivity analysis has shown that the savings order addition is preferred when the floor switch time is less than A (Figure 1), otherwise the floor savings order addition is preferred.

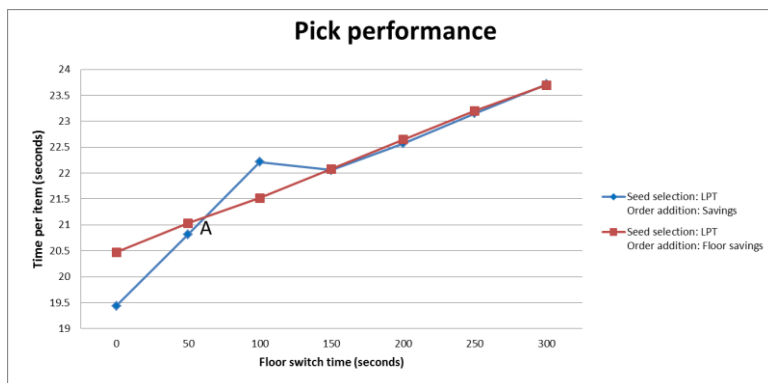


Figure 1 Trade-off of order addition rules on floor switch time

Overall, it can be recommended to Docdata to use the LPT seed selection and the floor savings order addition. The floor savings order addition, in which no additional floors are added to the seed, is chosen since the floor switch time is 190 seconds, which can be considered high.

Theoretical contribution

To our best knowledge, current literature is restricted to batch picking on a single floor; however, many warehouses, like the warehouse of Docdata, nowadays have multiple floors. Therefore, the effect of dividing a warehouse in multiple floors on pick performance is evaluated in this research. Sensitivity analysis has shown that an increased number of floors can be beneficial for the pick performance when the floor switch time is low in comparison to the number of cross aisles. On the other hand, locating all pick locations on one floor is more beneficial when the floor switch time is high in comparison to the number of cross aisles. Given the number of cross aisles and the floor switch time at Docdata, a single floor warehouse in which all aisles are located in succession performs better than a multiple floor warehouse in which the clusters of aisles are located above each other. This is shown in Figure 2.

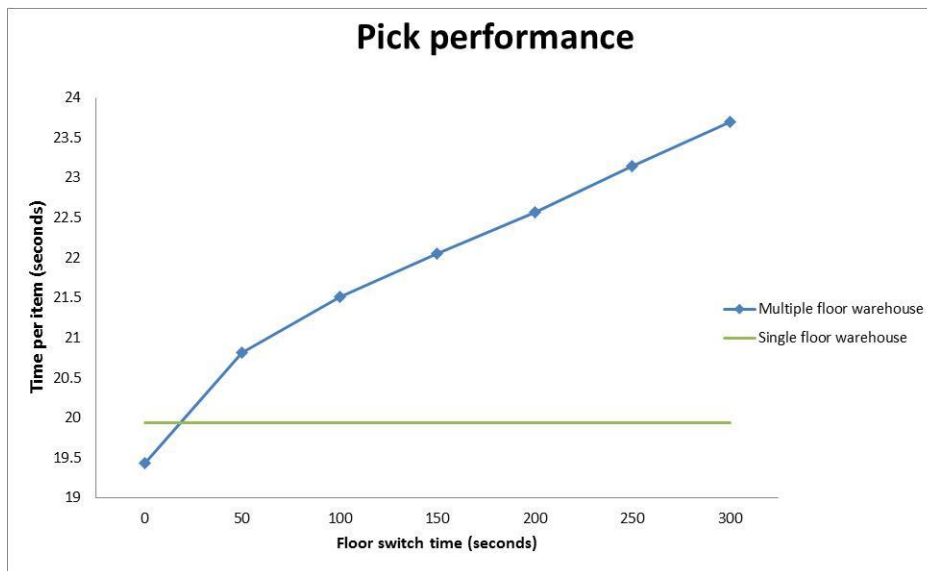


Figure 2 Pick performance in a multiple floor and single floor warehouse

Practical contribution

Practical usefulness of the project lies in the possible costs savings and service improvements that can be achieved by changing the releasing and batching strategy. As concluded in this research, when batches are only created when a picker is available, the LPT seed selection and the floor savings order addition are applied 17.5 per cent reduction in picking time can be achieved. Moreover, the due date performance can be improved from 99.4 to 100 per cent.

In addition, when the warehouse of Docdata is extended in future, it is recommended to make a trade-off between the floor switch time and the number of cross aisles in order to decide whether a single floor or a multiple floor warehouse is preferred.

Preface

This report is the result of my Master's thesis project, which I conducted in partial fulfilment of a Master of Science degree in Operations Management and Logistics at Eindhoven University of Technology. I performed this project at Docdata N.V. from April 2014 until August 2014. With submitting this thesis, my student life has come to an end. I experienced my time at university as really educational and enjoyable. Finishing my studies is not something I reached all by myself and therefore, I would like to thank some people.

I would like to thank my university supervisors. First of all, I want to thank Rob Broekmeulen for sharing his enthusiasm for his research domain. I have really enjoyed the discussion we have had on my research topic and really valued his founded feedback, devoted guidance and support during my project. It was a pleasure to work with him. I have learned a lot and I really appreciated his helpfulness and fast response during the entire project. In addition, I would like to thank Simme Douwe Flapper for sharing his knowledge and his clear feedback.

In addition, I would like to thank my company supervisors. In this regard, I would like to thank Docdata for providing me the opportunity to perform my research project. Especially, I would like to thank Simon Hooglugt, my company supervisor, for all the time and effort he put into helping and guiding me during the project. Moreover, I would also like to thank all my colleagues at Docdata, but especially Harm Dankers, Edwin Roovers and Bram van Beijnen, for their help, feedback and discussion.

Special thanks go to my parents, my boyfriend and my friends. First of all, thanks to my parents who enabled me to study. Moreover, thank you for your love and your support. For my boyfriend, thank you for your support, you always placed my effort on this thesis in the right perception. Last but not least I want to thank my friends. Thank you for the fantastic moments and the precious memories we have made: you have made my time as a student unforgettable. I enjoyed every piece of it and I would not have missed one single moment.

Anniek Verschure

Eindhoven, August 2014

Contents

Abstract	III
Executive summary	IV
Preface	VII
1. Introduction	1
1.1 Company description	1
1.1.1 Scope	2
1.2 Report structure	2
2. Methodology	3
3. Literature review	5
3.1 Pick routing	5
3.2 Batching	5
3.2.1 Zoning	7
3.2.2 Release moment of a batch	8
3.3 Literature gaps	8
4. Detailed analysis	9
4.1 The warehouse	9
4.2 Order fulfillment process	10
4.2.1 Order import	10
4.2.2 Order run	10
4.2.3 Batch creation	11
4.2.3.1 Release methodology	11
4.2.3.2 Batch clustering	11
4.2.4 Pick routing	12
4.2.4.1 Zoning	13
4.2.4.2 Depot location	13
4.2.5 Sorting and Packing	13
4.2.6 Sorting per zip code	13
4.3 Throughput times	13
4.4 Process scope	15
4.5 Travel distance	15
4.6 Time distribution order picking	17
4.7 Order patterns	18

5.	Problem description	20
6.	Research assignment	23
7.	Design	25
7.1	Assumptions	25
7.2	Key performance indicators	26
7.3	Experimental setting	27
7.3.1	Pick routing	27
7.3.2	Batch clustering	27
7.3.3	Release of batches	28
7.4	Scenario analysis	29
7.4.1	Degrees of freedom	29
7.4.1.1	Wave selection criterion	30
7.4.1.2	Seed selection	31
7.4.1.3	Order addition	31
7.4.2	Definition of scenarios	33
8.	Simulation model	34
8.1	Description of the used data	34
8.2	Validation and verification of the calculations	34
8.3	Parameter settings	34
9.	Results	38
9.1	Batch clustering	38
9.1.1	Seed selection	39
9.1.2	Order addition	40
9.2	Comparison with current situation	41
9.2.1	Practical usefulness	42
10.	Sensitivity analysis	43
10.1	Trade-off between savings and floor savings order addition	43
10.2	Effect of multiple floors	44
10.3	Effect of the S-shape strategy	46
11.	Conclusion and discussion	47
11.1	Theoretical contributions	48
11.2	Practical contributions	48
12.	Limitations and recommendations for further research	49

13.	Glossary	51
14.	Notation	55
14.1	Variables	55
14.2	List of abbreviations	55
15.	Literature	56

1. Introduction

This research is performed at Docdata N.V. and concerns the order picking and batching process. (Order) picking can be defined as the process of retrieving products from storage in response to a specific customer request (De Koster, Le-Duc and Roodbergen, 2007). Order batching is grouping a set of orders into a number of subsets, each of which can then be retrieved by a single picking tour (De Koster et al., 2006). This project is initiated because the numbers of items per order are increasing in recent years, while the current batch methodology only takes into account the location of one item of the order. Due to the limited information on the locations of the items in an order, inefficient batches arise. In addition, the picking process is the most labor-intensive operation in warehouses with manual systems and accounts for 55 per cent of total warehouse operating expense according to De Koster et al. (2007). It is therefore worthwhile to analyze and eventually redesign the order picking process at Docdata NV. Therefore, Docdata aims to get more insight into the performance of the picking process and ultimately wants to improve the simplistic batch methodology. The picking process can be decomposed in three hierarchical processes: the routing methodology, the batch clustering and the release moment of the batch. The aim of this research is thus to analyze and redesign those three processes. The objective function of the research is that all orders are handled in time, and that the time devoted to picking is minimized.

This paragraph starts with a description of the company. Then, the scope is delineated. The introductory paragraph concludes with the report structure.

1.1 Company description

Docdata NV is composed of four divisions: Docdata commerce BV, Docdata fulfillment BV, Docdata payments BV and IAI Industrial systems BV. Docdata commerce BV designs web shops for customers. Docdata fulfillment BV provides the logistic services for the client. The division Docdata payments BV provides the payments between customer and Docdata NV. IAI Industrial systems BV is a high-tech system builder. This research is conducted at Docdata fulfillment BV, from now on denoted as Docdata.

Docdata is a logistic service provider (LSP) which takes care of transport, storage, handling, packaging, returns and value added services of products sold online. These services are provided for clients like Bol.com, Zalando, Bijenkorf and V&D. Docdata operates in the e-commerce sector. In the e-commerce sector there are a high number of order transactions, order sizes tend to be small and there exist a high probability of significant fluctuation in customer demand (Tarn, Razi, Wen and Perez, 2003). Furthermore, the product assortment of a web shop is quite large.

Docdata has multiple warehouses in Europe, where products for different companies are stored and handled. The warehouse of Docdata is divided into multiple floors. Docdata performs the warehousing for companies in different sectors consisting of consumer electronics, fashion, games, books, furniture, cosmetics, CD/DVD/Blu-ray, toys, jewelry and finance.

1.1.1 Scope

Docdata provides logistic solutions for many clients who sell different items and offer different services to clients and thus require tailored designs. Because the processes for the various clients encompass differences, this project is executed for only one client. The largest client of Docdata is selected since an improvement in the efficiency of this client probably results in the largest time saving and due date improvement. In addition, Docdata has multiple warehouses where they provide logistic services. However, the main activities take place on the Veerweg location in Waalwijk in hall 1 and hall 4. Therefore, only those locations are in scope of this project. The existing layout of the warehouse is taken for granted. Moreover, the sequence in which items are stacked in the racks, the put-away strategy, while affecting the picking performance, are not considered. The warehouse of Docdata has roughly two functions: the major part of the orders is retrieved from stock and a minor part of the orders is retrieved cross dock. This research only considers the orders that are retrieved from stock.

1.2 Report structure

The report is structured as follows. First, the methodology of this research is described in Chapter 2. Then, a literature review is provided in Chapter 3. In this section an extensive review of existing literature is made and there is pinpointed which gaps are still left open. Thereafter, a detailed analysis is conducted in Chapter 4. In this analysis, the process which is fulfilled at Docdata is described, the warehouse layout is discussed, the throughput times are specified, and the travel distance and the time distribution of the current batches are given. Thereafter, the problem is defined in Chapter 5. Afterwards, the research assignment is explained in Chapter 6. Next, the design is formulated in Chapter 7, which encompasses assumptions, the key performance indicators, the experimental setting and a scenario analysis. Then the simulation model is explained in Chapter 8, including a data description and the parameter settings for the various scenarios. Subsequently, the results of the scenario analysis are given in Chapter 9. Subsequently, a sensitivity analysis is performed in Chapter 10, to pinpoint the effect of dividing the warehouse into multiple floors on the pick performance. This report is completed with a conclusion and discussion in Chapter 11 and recommendations for further research are described in Chapter 12. In Chapter 13 and 14 the used concepts and abbreviations are described.

2. Methodology

The aim of this research is analyzing the current picking and batching process and designing a new picking and batching process which reduces the time devoted to picking. As is graphically shown in Figure 3, this research can be roughly divided into three parts:

- Detailed analysis;
- Design;
- Testing the design.

The first stage of the research is devoted to the *detailed analysis*. In this part the order fulfillment process is analyzed. Therefore, four methodologies are applied: a field study, unstructured interviews with pickers, operators and logistic engineers, a data analysis and a literature review. The field study includes experiencing the process myself. A field study is chosen to experience possible inefficiencies. Unstructured interviews are chosen since the required information is at the pickers, operators and logistic engineers. The interviewed party has the possibility to show their story without being restricted to preset questions with this technique. Furthermore, operators and pickers are chosen because they daily encompass the problems occurring with the existing methodology. In addition, the logistic engineers are chosen because they have designed the current methodologies. The data analysis is chosen to be able to quantify the current performance of the methodologies. Due to computational restrictions, a fixed data set is chosen. The literature review is done to be able to use academic findings in a real industrial problem. Furthermore, by defining the gaps in existing literature, this research can contribute to existing literature. The detailed analysis results in a delineation of the scope and description of the problem.

In the second stage, the *design*, a redesign of the processes which are depicted in the problem description is made. The design is inspired by the data set acquired in the detailed analysis and literature in the field. The design is made by first defining the research assignment. Thereafter, the key performance indicators of the design are formulated. Given the problem definition, seven scenarios are defined.

The third stage, *testing the design*, is executed to evaluate the performance of the defined scenarios. In order to make a fair comparison possible, the parameter values should be determined. Because the performance of the scenarios is strongly dependent upon the parameter settings, the parameters are determined using a trial simulation run. Thereafter, the different scenarios are simulated on a fixed data set using Microsoft Access. Simulation is used because it is common in literature to assess warehouse performance. In addition, a simulation model offers the possibility to include realistic industrial aspects of the process, in contrast to a strict mathematical analysis which requires simplification of the research setting investigated. The data set is not treated deterministically, because this does not give a realistic picture of the situation. Instead, because the incoming time of orders is known, the data is treated dynamically. This means that during the day, information becomes available and decisions during the day can only be made based on available information. Then a sensitivity analysis is conducted, to identify the effect of multiple floors in a warehouse and to assess how sensitive the performance is to a change in the parameter values. Finally, the results are

analyzed and a conclusion is drawn, based on the key performance indicators of the scenarios and the sensitivity analysis.

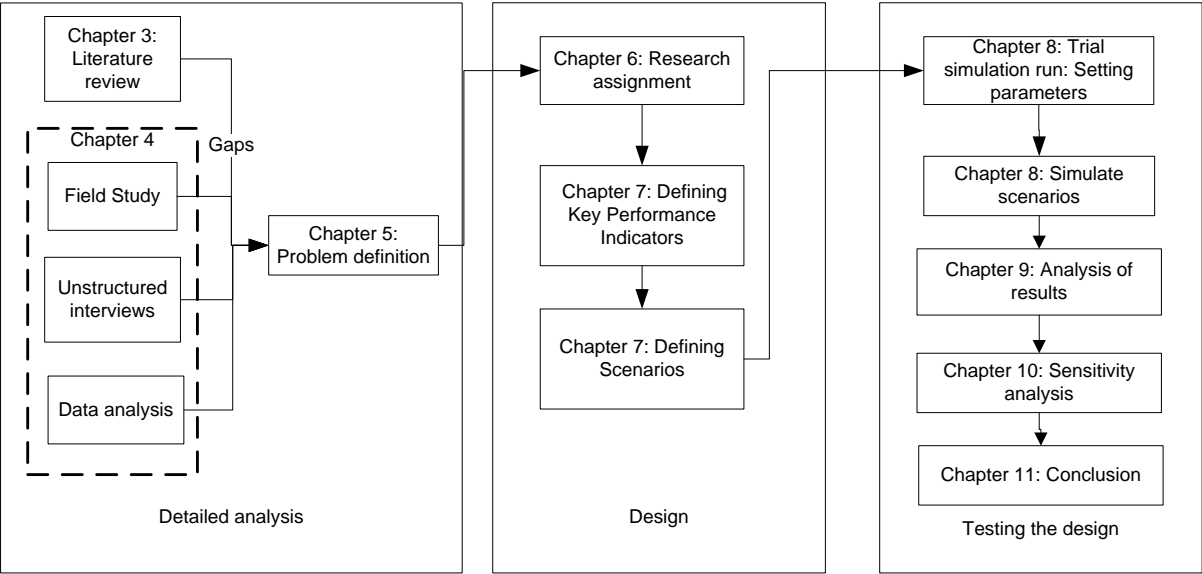


Figure 3 Methodology

3. Literature review

According to Xu, Liu, Li and Dong (2013) the main factors which impact order picking performance are the layout of the warehouse, the storage strategy, the routing policy, the zoning method and the batching policy. As noted in the scope, the warehouse layout and the storage assignment are outside the scope of this research; the other factors are discussed subsequently below. Afterwards, the gaps in existing literature are described.

3.1 Pick routing

Order picking is the process of retrieving products from storage in response to a specific customer requests and is the most labor-intensive operation in warehouses with manual systems (De Koster et al., 2007). The aim is to obtain the right amount of the right products for a set of customer orders (De Koster et al., 2007).

Two types of order picking systems can be distinguished: *parts-to-picker* systems and *picker-to-parts* systems (De Koster et al., 2007). *Parts-to-picker* systems are automated storage and retrieval systems, using mostly aisle-bound cranes to retrieve one or more unit load, and bring them to the pick position. In *picker-to-parts* systems, order pickers walk through the aisles to retrieve the orders (Dekker, De Koster, Roodbergen and Van Kalleveen, 2004). Two types of picker-to-parts systems can be distinguished: low-level and high-level order-picking systems. In a low-level order-picking system, the order picker picks the requested items from the storage racks, while travelling along the aisles (De Koster et al., 2006). In a high level systems, order pickers pick the items using a lifting order-pick truck or crane.

An important aspect of picking, strongly influencing the required time to pick a batch, is routing. According to Roodbergen (2001), routing policies can be organized according to two categories: optimal solutions and heuristics. Optimal solutions determine the shortest route of all possibilities. Heuristics are used to determine a feasible route, which is not necessarily the shortest one. Both approaches have their advantages and disadvantages. Heuristics are fairly simple in structure and therefore easy to implement in practical situations, but do not necessarily give the optimal solution (Roodbergen and De Koster, 2001). Optimal solutions, on the other hand, require larger calculation time (De Koster and Van Der Poort, 1998). De Koster and Van Der Poort (1998) found that the relative advantage of optimal solutions decreases when the number of picks per aisle increases. When there is chosen for heuristics, the most appropriate routing methodology is strongly dependent on the number of picks per aisle. De Koster, Van Der Poort and Wolters (1999) distinguish two routing strategy heuristics: the largest gap and S-shape strategy. In the largest gap strategy all aisles except the first and last visited are left at the same side as they were entered. This logic is given graphically in Figure 31 Appendix B. In the S-shape strategy the order picker enters every aisle where an item has to be picked and travels the entire aisle. This logic is given graphically in Figure 32 Appendix B.

3.2 Batching

Order sizes tend to be small in the e-commerce sector (Tarn et al. 2003); therefore, batching is often applied. Order batching is grouping a set of orders into a number of subsets, each of which can then

be retrieved by a single picking tour (De Koster et al., 2006). According to De Koster et al. (1999) substantial savings can be obtained by using well-formed batches. However, batch picking has the drawback that the units have to be sorted per customer after the picking process. In case that a large number of customers are batched at once, the process is not a simple job, and may even require an expensive sorting machine (De Koster et al., 1999).

Order batching can be classified into two types: static batching and dynamic batching (Xu et al., 2013). In static batching, the order information is known before batching. The problem is to decide which order should be assigned to which batch. In dynamic batching, the order information is not known before batching. This type of batching takes the stochastic property of the customer order into consideration. The problem is thus to determine the batch size or batch time window to minimize the expected throughput time of an arbitrary order.

De Koster et al. (1999) distinguish two types of batching algorithms that minimize the time per pick: *seed* and *savings* algorithms. Seed algorithms consist of two steps. First, the seed (or initial) order has to be selected from those orders not yet added to a route. Second, not yet selected orders have to be added to the seed order until the order picker is filled to capacity. Savings algorithms compare the saving in time s_{ij} they can obtain by combining two orders i and j in one route (with pick time t_{ij}) compared to the situation where both orders are collected individual (with order pick time t_i+t_j). Hence, $s_{ij} = t_i + t_j - t_{ij}$.

Under the assumptions¹, seed algorithms are best when the S-shape routing strategy is applied and the capacity of the pick device is large (De Koster et al., 1999). Under the same assumptions, savings algorithms perform better when the largest gap strategy is applied and the pick device has a small capacity. Thus, before a batching heuristic is chosen, first the routing strategy has to be selected. Therefore, the batch clustering methodology also depends on the routing methodology. It is known from Hall (1993) that S-shape is preferred over largest gap if the number of picks per aisle exceeds 3.8.

In order to apply a *seed algorithm*, the seed selection rule and the seed order addition rule have to be determined. All seed selection rules can be applied in a single mode, where the order is selected only once, and cumulative mode, where the seed order is renewed every time an order has been added to the route. De Koster et al. (1999) have found that the cumulative seed rule hardly performs better than the single-seed rule. De Koster et al. (1999) simulated six seed selection rules, and

¹ 1. Vertical movement of order pickers is disregarded.

2. Within an aisle, two-sided order picking is performed, that is simultaneous picking from the right and left side within an aisle.

3. Order sizes are generated from a discrete uniform distribution.

4. All information about orders to be picked is known in advance.

5. No orders may be spread over more than one batch, therefore the maximum order size should not exceed the pickers capacity.

6. All items of all orders have an equal weight as compared with picker's capacity.

7. There is a fixed depot.

compared these on distance. When seed algorithms are preferred, De Koster et al. (1999) have found that the following seed selection rules perform the best on time devoted to picking under the assumptions mentioned in the Footnote 1:

- The order with the largest number of aisles that have to be visited;
- The order with the longest travel time;
- The order with the largest aisle range (absolute difference between the most left aisle number and the most right aisle number to be visited).

If the seed selection rule is selected, the seed order addition has to be determined. De Koster et al. (1999) have shown that when the S-shape routing strategy is applied, the following seed order addition rule is preferred under the assumptions mentioned in the Footnote 1:

- Choose the order that minimizes the number of *additional aisles*, compared to the seed order, that have to be visited in the route.

3.2.1 Zoning

A batch can be picked in full, or can be divided into zones. According to De Koster, et al. (2006) a combination of batching and zoning can significantly increase productivity. In order to pick a batch in zones, the total picking area is divided into zones, each order picker is assigned to pick the part of the order that is in his assigned zone (Koster et al., 2006). Advantages of zoning include that each order picker needs to traverse only a smaller area, reduced traffic congestion and the fact that order pickers become familiar with the item locations in the zone. However, the main disadvantage is that orders are split and must be consolidated before shipment to the customer. According to Parikh and Meller (2008) the decision between batch and zone picking depends on four factors: pick-rate, aisle blocking, workload-imbalance and sorting. When zone picking is applied, the pick-rate will increase and aisle blocking is eliminated. However, the work imbalance increases and additional sorting activities are needed.

Roughly, two types of zoning can be distinguished: parallel (or synchronized) picking and progressive assembly of an order (De Koster et al., 2006). In parallel picking, a number of pickers start on the same order, each order picker in his own zone. The partial orders are merged after picking. The second approach, progressive assembly, uses one order picker to start on the order and when he finishes his part, it is handled to the next picker. An important issue in progressive zoning, especially when the demand has a high variance, is that the workload must be equally distributed over the order pickers. An alternative for progressive zoning with fixed zone sizes would be to size the zones and assign the order pickers dynamically to the zones. An example of this is the bucket-brigades concept (De Koster et al., 2006). In this concept, all order pickers are working on separate batches and if the order picker closest to the end of the line deposits his finished batch, he walks back along the line towards the starting point. If he meets another order picker, he takes over the batch from the other person and continues picking this batch. The order picker from which the batch was taken moves back along the line until he meets another order picker, and so on. Thus, one order picker starts all orders. The main advantage of bucket brigades is that the workload of the pickers is balanced (De Koster et al., 2006).

3.2.2 Release moment of a batch

One of the most important problems in warehousing in the e-commerce is the availability of the retrieved orders at the right time. Three types of release moment decisions exist, *time-based* schemes, *quantity-based* schemes and *quantity-time-based* schemes (Chen, Wang and Xu, 2005). Under a time-based scheme, a release is dispatched periodically. Under the quantity-based scheme, a release is dispatched when a certain quantity of outstanding demand is accumulated. The *time and quantity based scheme* is the hybrid form of the previous mentioned. In this scheme, all orders are on hold until one of the following situations occurs: a predetermined shipping date or an accumulation of a prefixed quantity. Chen et al. (2005) found that the *quantity-based scheme* often outperforms its time-based counterpart, and no exception is found in which this is not the case. However, a drawback is that the quantity based scheme needs continuous monitoring.

Elsayed and Lee (1996) take the due date of an order when constructing batches into consideration. The objective of the construction is to minimize the tardiness of the order. Where the tardiness of order i is defined as $\max(Completion\ time_{order\ i} - Due\ date_{order\ i}; 0)$ (Elsayed and Lee, 1996). They rank the orders on due date in non-decreasing order, and then propose three rules to minimize the tardiness: the nearest schedule (NS) rule, the shortest processing time (SPT) rule and the most common locations (MCL) rule. The NS and SPT rules show better measures than MCL rules as the tightness of the problem increases. Thus, when the due date is tight, it is better to process the orders as fast as possible.

Pang and Muyldermans (2013) studied the impact of postponing routing decisions on the travel distance for the capacitated vehicle routing problem (CVRP). The CVRP is the problem of finding a set of minimum cost routes visiting clients with non-negative demands and such that: each route starts and ends at the depot, each client is visited exactly once, and the total load on each route does not exceed the vehicle capacity (Pang and Muyldermans, 2013). They assume that orders come in during the day and that the service provider can decide when to visit and service the clients. Their results reveal that postponement strategies can reduce the travel distance significantly; however, the benefits depend on the specific routing context. The benefits from postponement are larger when the client density is lower, the vehicle capacity is larger and when the depot is more centrally located in the service area.

3.3 Literature gaps

Literature on batch picking is available in great abundance. However, some possibilities to contribute to current literature still exist. To our best knowledge, current literature is restricted to batch picking on a single floor. However, many warehouses like the warehouse of Docdata, have multiple floors. Vertical movement cannot be disregarded, when a warehouse is divided into multiple floors. This research contributes by designing methodologies for multiple floor batch picking and by pinpointing the effect of multiple floors on the pick performance.

4. Detailed analysis

This chapter describes the current situation at Docdata. The information is acquired by a field study, unstructured interviews and data analysis. Unless stated otherwise, all numerical examples are based on the non-peak week 11 in 2014.

This chapter entails an explanation of the warehouse in study, a process description, the throughput times of the process, the travel distance of the picking department, the time distribution of the picking department and an analysis of the order patterns.

4.1 The warehouse

The warehouse in scope of this research is divided into two halls: hall 1 and hall 4. These halls are divided into catwalk and pallet locations as shown in Figure 4.

Catwalk Hall 4	Pallet Hall 4
Catwalk Hall 1	Pallet Hall 1

Figure 4 Overhead view of the warehouse in study

A catwalk is defined as a storage area with multiple floors connected by elevators. The catwalk of hall 1 is divided into 5 floors, while the catwalk in hall 4 only uses the top floor. This is shown in Figure 5. The catwalk has a cross aisle, next to the normal aisles. On each floor, the picking area is rectangular (no unused space), and consists of a number of parallel aisles. The pallet areas are also rectangular, without cross aisles.

Elevator	Floor 4	Floor 4	Elevator
	Floor 3		
	Floor 2		
	Floor 1		
	Floor 0		
	Hall 1	Hall 4	

Figure 5 Frontal view catwalk

Next to the catwalk floors and pallet areas, two separate zones are defined for large releases. In total, ten zones can be distinguished.

4.2 Order fulfillment process

When an order, consisting of single or multiple item(s), is handed over to Docdata, it will go through seven phases as shown in Figure 6. In the order import, the orders are placed on the server of Docdata. Then, the order run releases the orders which have to be handled today. Thereafter, the orders are placed in batches, and the pick routing determines in which order the batch have to be picked. Subsequently the batches are disaggregated to orders again in the sorting process. For distribution purposes, the orders are packaged and sorted per zip code. Thereafter, the order is handed over to an external distribution company. A more detailed process description is given in this chapter.

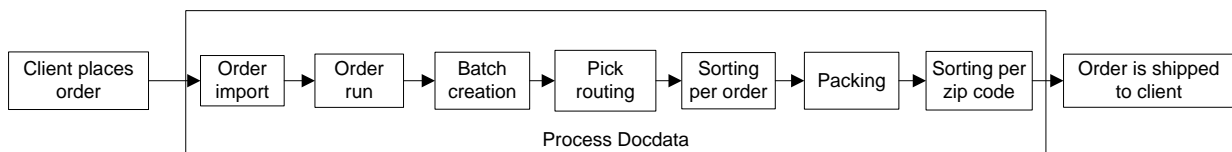


Figure 6 The sequence of processes

4.2.1 Order import

During the order import process, orders placed in the web shop of the client are communicated to Docdata. When an order is communicated before 11 PM on weekdays and the system displays stock availability, the order should be handed over to the external distributor before 1 AM. When the system does not display stock, the delivery date is extended. In addition, customers who place an order in a web shop can also choose for a later delivery date themselves.

4.2.2 Order run

During the order run a check is performed whether the hold date equals today and if the items are in stock. The hold date is defined as the date that the order has to be handed over to the distributor. In order to assess whether the order needs to be processed today, three types of orders can be distinguished. When an order consists of multiple items with different hold dates, then the earliest hold date is taken into consideration.

- A must-go-order: an order of which the hold date is less than or equal to today.
- A can-go-order: an order of which the hold date is later than today.
- A 24h order: an order that has to be delivered in 24 hours. (Every 24h order is thus a must-go-order, but not vice versa.)

The different types of order runs which are distinguished in the system are given in Appendix C Paragraph 16.3.1. During an order run, orders that should be picked that day are released. Each released order contains all items required and their associated location(s). When an item is stored at multiple locations, the pick location is determined by the lowest inventory position and not by the shortest distance. The orders including a pick location serve as input for this project.

4.2.3 Batch creation

Batches should be made dynamically because the batch creation starts far before all information of incoming orders is known (Xu et al., 2014). Batch creation consists of two processes: first the orders are released by an operator and thereafter, the released orders are clustered to compose batches that will be retrieved simultaneously. These processes will be discussed in this chapter.

4.2.3.1 Release methodology

Docdata has agreed with the customer to process all orders before a certain cut-off time, the moment of batching is thus restricted by the cut-off time. Currently, a time and quantity based releasing policy is used. In this scheme, all orders are hold until the one of the following situations occurs: a predetermined date or an accumulation of a prefixed quantity (Chen et al., 2005).

The quantity based scheme releasing policy at Docdata is based on both the number of orders in queue in front of the outbound lines, and whether the pickers can still keep operating. At an outbound line the batches are sorted to orders and packed. Multiple outbound lines are available at Docdata. A description of all outbound lines is given in Appendix C Paragraph 16.3.4. The aim of the quantity based releasing policy is retaining a constant queue in front of the outbound line. Whether the pickers can keep operating is calculated with an average picking efficiency. However, the number of picks per person per hour is highly variable.

All due dates of the released orders are equal. A time based frame assures each order is handled within a predetermined time frame (Chen et al., 2005). Because Docdata has agreed to process all orders before a certain cut-off time, a time based frame is needed.

When a batch is released, the batch is not always picked immediately. There is no clear alignment between the batching moments and picking moments of the items. The time between batch creation and first pick, when eliminating all outliers above 12 hours, is on average 90 minutes and distributed as shown in Figure 34 Appendix B. When a picker starts its batch, the incoming orders of the last 90 minutes are not taken into consideration. There is thus an average information loss of 90 minutes. This loss is caused by a lack of information transparency. In addition, order pickers believe if they create fewer batches, pickers cannot keep operating. Thus in future, the batch creation moment has to be more aligned with the pick capacity. When the batches are released later, the number of available orders increases and thus enables the batch clustering to reduce the average travel distances per pick, and thereby increases the pick performance (Pang and Muyldermans, 2013).

4.2.3.2 Batch clustering

Orders consist of order lines: each line for a unique product or stock keeping unit (SKU) in a certain quantity. At Docdata wave picking is applied. The term wave picking is used if orders for a common destination are retrieved simultaneously (De Koster et al., 2006). A wave consists of all orders that are allowed to be retrieved or sorted together. A batch is a subset of orders in a wave, limited by the pick cart capacity. The pick cart capacity per wave is defined as the number of items that fit in the pick cart. The dependence of orders, batches and waves is given in Figure 7.

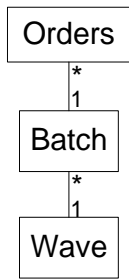


Figure 7 Dependence of orders, batches and waves

Per wave, the batches are composed by choosing a seed (initial) order and then adding orders until the pick device capacity is reached. The seed order is the order that has the lowest alphabetic location number on the first order line. The order addition is done on the lowest alphabetic location number on the first order line as well. This implies that, when orders are available with multiple order lines located at separate warehouse areas, the possibility exists that a picker may need to switch more warehouse areas than theoretically necessary. This is caused by the fact that only the information on the first order line is taken into consideration. Concretely this means that 10.75% of all orders are located at multiple warehouse areas, while 37.6% of all batches go over multiple warehouse areas. When multiple warehouse areas are needed, the distribution over the number of warehouse areas needed for the orders and batches is given in Figure 8. For the orders, the number of warehouse areas needed decreases in frequency of occurrence. The number of batches which traverse 2,3,4 or 5 warehouse areas is more or less constant. This means that the average number of warehouse areas per batch is nearly two, while the average number of warehouse areas per order equals circa 1,1.

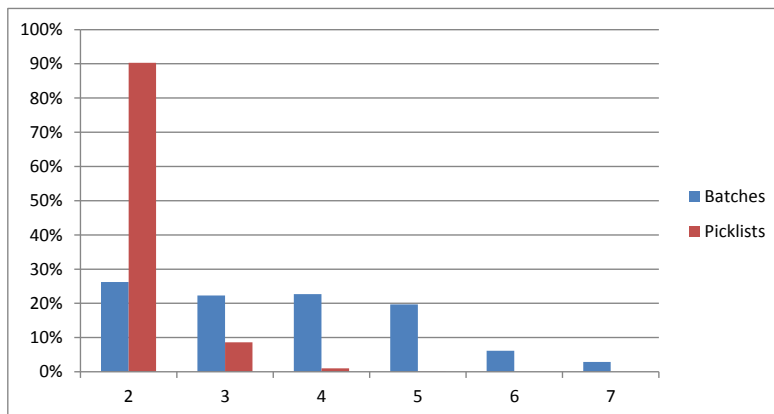


Figure 8 Average number of floors needed per order (pick list) and batch

4.2.4 Pick routing

The aim of determining order picking routes consists of finding a sequence in which products have to be retrieved from storage such that travel times are as short as possible (Roodbergen and De Koster, 2001). Docdata uses a picker-to-parts system (De Koster et al., 2006). At Docdata, order pickers are able to traverse aisles in both directions and can change direction within an aisle. Items are stored on both sides of the aisle. Although pickers can change direction within an aisle, the aisles are still relatively narrow, and therefore double sized order picking is applied. Items are picked per wave. For

every wave, the items can be stacked in the pick device in every sequence. The picking areas of the different waves overlap each other. Because of the overlap of the waves and the fact that batches are created multiple times a day, every aisle has to be traversed on average 55.5 times a day.

At Docdata the S-shape strategy (or traversal strategy) is applied. In this strategy, the order picker enters every aisle where an item has to be picked and travels the entire aisle. This logic is given graphically in Figure 31 Appendix B. In order to assess whether the S-shape strategy is an appropriate strategy, the picks per aisle should be determined. The picks per aisle are as shown in Table 2 Appendix A. Data analysis indicated that for four outbound lines the picks per aisle exceed 3.8 and thus the S-shape strategy is appropriate. For the other three outbound lines the picks per aisle are less and thus S-shape is not appropriate according to Hall (1993).

4.2.4.1 Zoning

At Docdata, progressive assembly is used. When a batch has to traverse multiple floors, for each floor a picker is allocated to the batch. The first picker picks all items on first floor, places the batch in front of the elevator and transposes the batch to another picker who picks all items requested on the second floor. Vertical distance takes considerably more time than horizontal distance. Therefore, this method increases the productivity of the pickers when compared to conventional batching in which the picker also switches the floors together with their pick device. After the pick device traversed the floor it has to be handed over to a next picker. However, the exact location of the batches is unknown by the system and therefore this is hard to manage.

The other form of zoning, parallel picking, is not feasible at Docdata because the sorting capacity is inadequate to accumulate the orders after picking.

4.2.4.2 Depot location

At Docdata every route has to start and end at a depot where pickers receive empty pick devices and batch forms and deliver full pick devices. There are two fixed depot locations at Docdata located at floor zero or floor two depending on which wave should be picked. The depots are located in the corner of the warehouse.

4.2.5 Sorting and Packing

The orders are picked in batches per outbound line, thus the batch has to be disaggregated later. This is done via the sorting process. Afterwards, the orders have to be packed. This is done via the packing process. The distribution of items per outbound line is given in Figure 35 Appendix B and Table 5 Appendix A.

4.2.6 Sorting per zip code

After the orders are packed, they have to be sorted per zip code. This is done by the POST sorter or PAKKET sorter. The orders have to be sorted before 1 AM, because then the distributor will take them over.

4.3 Throughput times

To reveal the importance of this research, the due dates and process times of the processes given in Figure 6, are described in this chapter. The client can place orders on the Docdata server 24 hours a

day. However, only orders that are placed before 11 PM are handled the same day by Docdata. Therefore, the last order run of the day is executed at 11 PM. In order to handle the daily orders, the order picking department starts generally at 4 PM with creating batches and order picking. If it is not possible to handle all orders from this time on the picking department starts earlier. This process should be finished before 00.30 AM, because the retrieved batches should be sorted and packed before 1 AM, because then the external distributor takes over the orders. These cut-off times are given in Figure 9.

Internal analysis reveals that the order import time is negligible. In addition, this research assumes that the time to release the orders by the order run and to create a batch is negligible. To calculate the throughput of the whole process, the process time of the order picking, sorting, packing and zip code sorting has to be calculated respectively. The order picking efficiency is dependent upon the number of pickers available.

Then the orders have to be sorted and packed. In order to calculate the process time of the machine a dataset from 01.06.2013 to 01.03.2014 is taken. Because the machines start operating after 5 PM and there is no break of the operators before 7 PM, the process time per order is averaged over this time slot. When multiple machines are used for the pack and sort process, the throughput of these machines is summed. Then the orders should be sorted per zip code. These processing times are also given in Figure 9.

As can be seen, all orders placed before 11 PM have to be picked, sorted, packaged and sorted per zip code before 1 AM, which is only two hours. As explained the time slot to process all orders is relatively tight and shortening the throughput time of the orders is thus worthwhile considering.

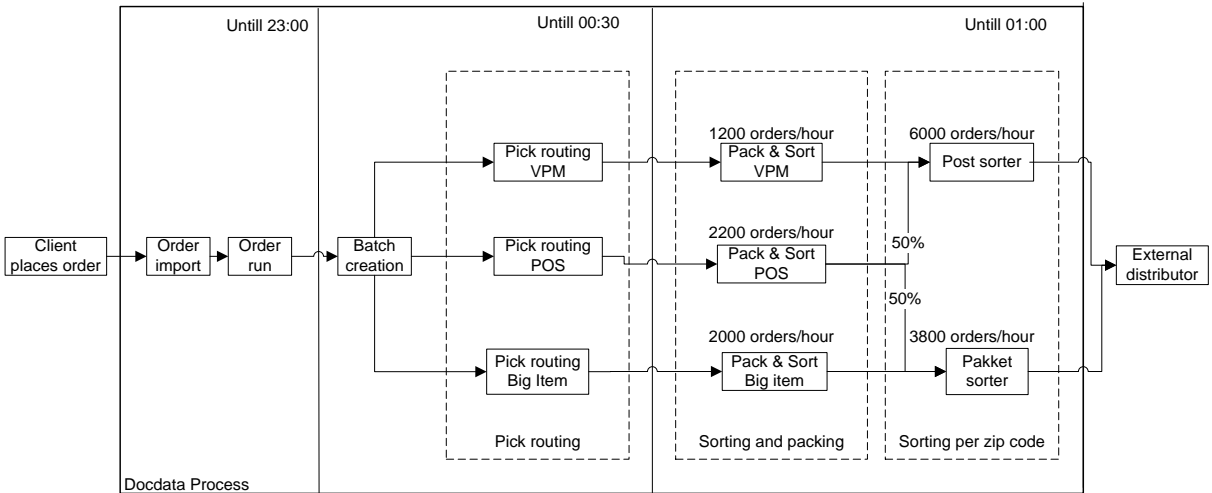


Figure 9 Throughput times

4.4 Process scope

Given the process description in Paragraph 4.2 and 4.3 a clear and feasible delineation of the scope of this research is described in this section. The first delineation is made on outbound lines; only the outbound lines POS, VPM and big items are considered. This means that orders which are picked with invoice, URO orders (defects) and NCO orders (retour supplier) are out of scope.

The major delineation is the process delineation given in Figure 10. Only the batch creation process and the pick routing are taken into consideration. When the sorting, packing and sorting per zip code is taken out of scope, the number of sorting directions (chutes) that the sorting machine has is taken for granted, and can be seen as functional requirement for the design. The VPM does not sort the orders since it packs only single item orders. The POS sorter has 271 sorting directions (chutes), meaning that the theoretically maximum number of orders that can be combined in a batch is 271. Currently, no more than 150 orders are combined in a batch and thus the number of chutes of the sorter does not form a restriction on the batch sizes. For the big item outbound line, the orders are large in terms of dimensions. Therefore the pick cart capacity for these waves is very restrictive. The sorting directions are larger than the number of orders that fit in a pick cart. When taking into account this maximum, the sorting, packing and sorting by zip code can be placed out of scope.

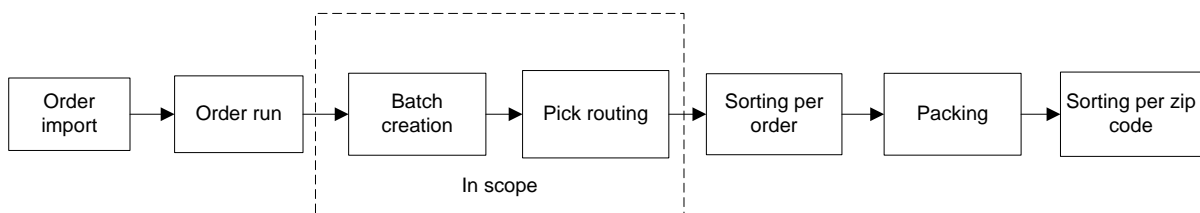


Figure 10 Process delineation

The solution that is provided is designed on a fixed set of data. Whether the solution fits the process, depends on the accuracy and representativity of the data.

4.5 Travel distance

According to Tompkins et al. (2003) 50 per cent of the picking time is devoted to travelling through the aisles. Therefore, it is important to calculate the travel time of the current batches. It is assumed that the velocity of an order picker is constant and hence the travel time is linear to the travel distance.

Each batch includes multiple pick locations $\{1, \dots, n\}$. Every pick location has to be visited exactly once, and the sequence in which the locations are visited is determined by the routing methodology and is assumed S-shape. In contrast to the distance approximations as given by Roodbergen and Vis (2006), products are not uniformly distributed over the aisles and locations, as shown in Figure 33 Appendix A. In addition, distance approximations in existing literature neglect vertical movements, while the warehouse of Docdata has multiple floors. Therefore, these approximations cannot be applied. As known from previous research, the total distance of a batch consists of the distance within the aisles, the distance between aisles and additional to research done so far, distance between floors. These distances will be explained and calculated in this chapter, the following notation is used:

Notation

a_f = Number of aisles on floor f

$a_{i,f}$ = Aisle of the i th pick on floor f

$da(l)$ = Aisle length of the location type l (Figure 11)

$dc(l)$ = Cross aisle distance of the location type l (Figure 11)

d_s = Total horizontal distance to pick all items of the orders in batch s

F = floor number, $f \in F, f = \{0,1,2,3,4\}$

f_s = Number of floors in batch s

$n_{s,l,f}$ = Number of visited aisles of batch s on location type l of floor number f

L = Location type, $l \in L, l =$

$\{\text{Catwalk hall 10, Catwalk hall 13, Pallet hall 10, Pallet hall 13, Release Canteen, Release Mezzanine}\}$

s_l = Start aisle of location type l : $s_l =$

$\{S_{\text{Catwalk hall 10}}, S_{\text{Catwalk hall 13}}, S_{\text{Pallet hall 10}}, S_{\text{Pallet hall 13}}, S_{\text{Release canteen}}, S_{\text{Release mezzanine}}\}$

tf = Floor switch time

v = Horizontal travel speed

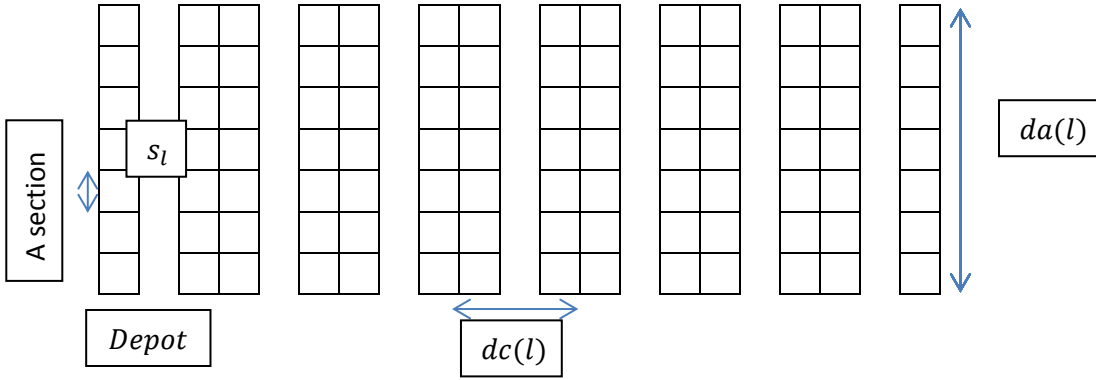


Figure 11 Explanation of in aisle and cross aisle distance

The in-aisle distance is the horizontal distance that an order picker walks through an aisle. Because Docdata uses an S-shape routing policy, every aisle which contains a pick should be traversed entirely. Given the data of week 11, every aisle contains at least one pick. Therefore, this research assumes that it is not possible to make the warehouses more compact. The length of an aisle is dependent on the area location in the warehouse. The aisle length is equal to a multiplication of the number of sections and the section width. The lengths are shown in Table 3 Appendix A. Double sided picking is applied at Docdata, but the distance to cross an aisle is assumed to be zero. This assumption is common in literature when the width of the aisle is narrow, which is the case at Docdata (Roodbergen & Vis, 2006). In order to calculate the in-aisle distance the number of aisles that should be traversed has to be multiplied by the aisle length. Because every route has to start and end at the depot, it is not possible to traverse an odd number of aisles. Therefore, the number of aisles needed is rounded up to the nearest even number.

$$\text{In - aisle distance} = 2 \cdot \sum_{l \in L} \sum_{f \in F} \left\lceil \frac{n_{s,l,f}}{2} \right\rceil \cdot da(l) \quad (1)$$

The cross-aisle distance is the distance needed to switch from one aisle to a subsequent aisle. The cross-aisle distance is dependent of the aisle width and the depth of the racks. Both, the aisle width and the depth of the racks differ for catwalk and pallet locations. Therefore, the cross-aisle distance for catwalk and pallet are different. The total cross-aisle distance per batch should be calculated per floor because the elevator is located next to the depot. The cross-aisle distance can be decomposed into two components: the distance between the depot and the head of the farthest aisle, and the distance backwards. The cross-aisle distance can therefore be computed by multiplying the number of aisles that need to be crossed by the associated cross-aisle distance. Symmetric distances are assumed, which means that the distance from x to y is equal to the distance from y to x . Since the distance is walked twice, the cross-aisle length should thus be multiplied by two.

$$\text{Cross - aisle distance} = 2 \cdot \sum_{l \in L} \sum_{f \in F} (dc(l) \cdot (\max(a_{i,f}) - s_l)) \quad (2)$$

Combining the previous formulas, the total horizontal distance for batch s , d_s , can be computed by (3).

$$d_s = 2 \cdot \sum_{l \in L} \sum_{f \in F} \left(\left\lceil \frac{n_{s,l,f}}{2} \right\rceil \cdot da(l) + dc(l) \cdot (\max(a_{i,f}) - s_l) \right) \quad (3)$$

It is assumed that the horizontal velocity of an order picker is constant and hence the horizontal travel time is linear to the horizontal travel distance. This is a common assumption in existing literature (De Koster, 1999). The warehouse is composed of multiple floors and therefore the time to switch floors should be determined as well. The floor switch time is dependent upon the waiting time in front of the elevator, and the actual vertical movement. The actual vertical movement is a minor portion of the floor switch time, and therefore the floor switch time is assumed independent of the combinations of floors. Furthermore, the time to switch a floor is assumed independent of the composition and size of a batch and hence assumed constant. The total time needed to pick batch s , is given in formula (4).

$$\text{Travel time needed for batch } s = \frac{d_s}{v} + tf \cdot f_s \quad (4)$$

4.6 Time distribution order picking

The total time devoted to picking at Docdata in 2014 is expected to be equal to 132.500 hours. Because the sales are expected to grow, it is expected that the total pick time will increase as well. In this paragraph, the time needed to retrieve all orders is defined. Therefore, additional to the notation mentioned in Paragraph 4.4, the following notation is used.

Notation

et = Extraction time per item (seconds)

i_s = Number of items in batch s

S = the set of all feasible batches, $s \in S$

st = Startup time of a batch

The time devoted to a batch can be divided into startup time, extraction time and travel time. The startup time includes pick up an empty pick device and a printed batch list, this has to be done for all types of batches. Therefore, it is assumed that the startup time of a batch is independent on the composition and size of the batch. This implies that the total starting up time is constant and only dependent upon the number of batches executed during the day. The extraction times include pick and search activities and are assumed to be independent of the composition of a batch. This assumption implies that the total extraction time is a constant per item, irrespective of the batch composition and actual routing through the warehouse. The travel time is already defined in Paragraph 4.4. Using this information, the time devoted to picking is given in formula (5).

$$\text{Time devoted to picking} = \sum_{s \in S} \left(\frac{d_s}{v} + tf \cdot f_s + st + i_s \cdot et \right) \quad (5)$$

An internal time and motion study denotes the travel speed is 0.9 m/s and data analysis reveals that the startup time per batch is equal to 146 seconds, and the floor switch time is equal to 190 seconds. Using the average pick time per item, the distance per pick, the constant startup time and the constant floor switch time, the extraction time per item can be calculated and is equal to 7.7 seconds. Using this information, the time distribution can be acquired and is given in Figure 12 can be acquired. It is shown that, 75% of the picking time is devoted to travel time, and therefore worthwhile optimizing.

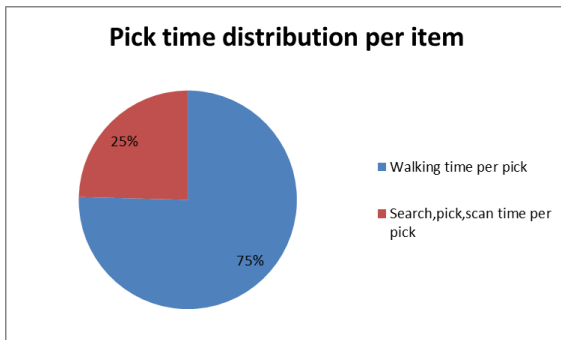


Figure 12 Pick time distribution per item

4.7 Order patterns

As the total travel time function shows, the travel time of an order or cluster of orders is only composed of the number of aisles ($n_{s,l,f}$), the number of floors (f_s) and the distance between the farthest aisle and the depot ($\max(a_{i,f}) - s_l$). Thus the combination of aisles and floors determine the distance of an order. When combining two orders with the same order and floor combination, this will not result in extra distance. Whether orders have similar aisle and floor combinations should be assessed. Every unique combination of aisles in a floor can be given a unique number. The possible unique combinations per floor is equal to $2^{\text{number of aisles per floor}}$. When orders that are located over multiple floors are neglected, the combinations over multiple floors can be summed. For the existing layout, this means that $13.5 \cdot 10^7$ floor and aisle combinations are possible. However, data analysis has revealed that the order pattern results in significant less aisle and floor patterns (3281). When we compare the number of prevailing patterns with the number of orders, Figure 13 can be acquired. As Figure 13 shows, the order patterns are rather similar at Docdata.

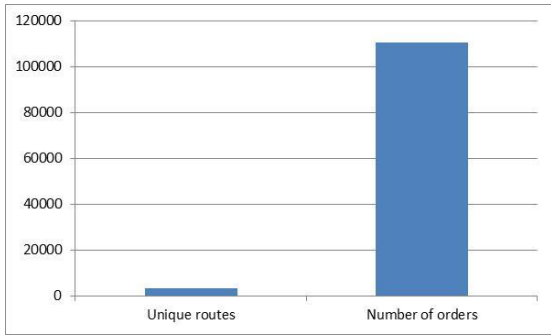


Figure 13 Unique aisle and floor combinations

5. Problem description

The main objective of the research is to minimize the time devoted to picking, which ultimately leads to increased picking efficiency. As explained in Paragraph 4.5 and 4.6, it is assumed that the extraction time of items and velocity of pickers is constant. Furthermore, it is assumed in paragraph 4.6 that the total start up time is only dependent on the number of batches executed during the day. Therefore, when improving picking efficiency the travel distance, the required number of floor switches and the number of batches are the only aspects which can be influenced in this research.

The main problem can be decomposed in three hierarchical problems: the routing methodology, the batch clustering and the release moment of the batch, as shown in Figure 14. The design made in one phase of the hierarchy strongly influences the decisions to be made in the other phases of the hierarchy. The decision made in the top of the hierarchy reduces the options for the decisions made in the lower level. However, the lower level decision has also impact on the higher level decisions of the hierarchy, on these decisions should be anticipated as mentioned in Schneeweiss (2003). Therefore, the problems cannot be seen in isolation. Since the decisions made on higher hierarchical levels reduce the design options in the lower levels, the design is made bottom up.

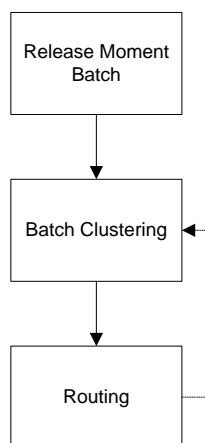


Figure 14 Problem hierarchy

Routing, the problem in the bottom of the hierarchy, is determining in which sequence the items of a given batch should be retrieved. Thus, given that a cluster of items should be retrieved at given locations, in what sequence should the order picker visit these locations to minimize travel distance? Preferably, every aisle has to be traversed only once and therefore all picks in an aisle should be retrieved simultaneously. Hall (1993) showed that the S-shape strategy is preferred when the numbers of picks per aisle exceed 3.8. Then the question remains how many picks per aisle are available. The batch clustering in line with the demand pattern and put away strategy, determine the number of picks per aisle.

The picking process becomes more efficient when a combination of orders, a batch, is picked rather than the orders are picked per order (De Koster et al., 1999). However, when batch picking is applied, the orders have to be sorted. This is not a simple process, and may require a sorting machine (De Koster et al., 1999). Because such a sorting machine is available at Docdata, batch picking results in

additional savings compared to picking per order. Given that batch picking is applied, the problem that remains is how the orders should be clustered such that the total picking time is minimized.

Given is that every route has to start and end at the depot and every order has to be retrieved. In addition, the pick device in which the orders are picked has a limited capacity, and each route cannot exceed the pick device capacity and thus the batching problem is a *capacitated vehicle routing problem* (Fisher, Jaikumar and Wassenhove, 1986), (Pang and Muyldermans, 2013). In addition, every tour starts and ends at the depot, where the order picker can deliver full bins and pick up empty bins. Docdata has 12 separate outbound lines, from which a route can start or end. Although there are 12 outbound lines, there are only two locations to start and end a batch, located at the corner of the warehouse. Therefore, it is a *multiple depot capacitated vehicle routing problem*.

This research considers a parallel-aisle warehouse connected by three cross aisles: in the back the front and the middle, where n orders need to be picked. The warehouse consists of multiple floors connected by an elevator. The orders are picked manually by order pickers that use a pick cart with a limited capacity dependent on the wave. Given the existing sorting capacities, it is not possible to accumulate the orders after picking. Therefore, all orders should be retrieved in its entirety and parallel picking is not possible. Furthermore, as mentioned in Paragraph 4.4, the extraction times are assumed constant, irrespective of the batch composition and actual routing through the warehouse. Accordingly, the extraction times may be omitted from the optimization problem altogether. This problem under study is proved to be NP-hard in the strong sense for more than three orders (Gademann and Van De Velde, 2005). In order to define the problem mathematically, next to the notation mentioned in Paragraph 4.5 and 4.6, the following notation is used.

Notation

$$a_{js} = \begin{cases} 1, & \text{if and only if order } j \text{ is included in batch } s \\ 0, & \text{Otherwise} \end{cases}$$

$c_w =$ maximum number of orders per batch dependent on the type of wave w

$$x_s = \begin{cases} 1, & \text{if batch } s \text{ is selected} \\ 0, & \text{Otherwise} \end{cases}$$

Mathematically, the problem is to determine values of decision variable x_s that minimize:

$$\sum_{s \in S} \left(\frac{d_s}{v} + tf \cdot f_s + st \right) x_s \quad (6)$$

Subject to:

$$\sum_{j=1}^n a_{js} \leq c_w \quad (7)$$

$$\sum_{s \in S} a_{js} x_s = 1, \text{ for } j = 1, \dots, n. \quad (8)$$

$$\forall s: a_{js} \leq x_s \quad (9)$$

$$\forall j: \sum_s a_{js} = 1 \quad (10)$$

$$x_s \in \{0,1\}, \text{ for each } s \in S \quad (11)$$

A batch is only feasible when it does not exceed the pick cart capacity per wave, therefore constraint (7) is defined. Constraint (8), (9), (10) and the integrality condition of constraint (11) ensure that each order is assigned to exactly one batch. The number of feasible batches, S is equal to $\binom{n}{c}$, the size of the LP problem increases significantly by increasing n but in particular when increasing batch size c (Gademann and Van De Velde, 2005). Due to the large amount of orders and the relatively high capacity of the pick device at Docdata the batching problem in study is not solvable in polynomial time (Gademann and Van De Velde, 2005), and therefore, a heuristic is chosen. An effective batch clustering methodology is dependent on a decision which lays lower in the hierarchy: the routing methodology. In literature, the most commonly used routing heuristics are the largest gap and S-shape strategies (Roodbergen & Vis, 2006), (De Koster et al., 1999). After the decision is made to route the orders largest gap or S-shape, the decision whether to cluster the batches according to the seed or savings algorithm should be made.

In addition, the maximum size of a batch has to be computed. The batch size is dependent on the capacity of the pick device per wave and the number of sorting directions (chutes) of the outbound line. This is a *bin packing problem* (Jansen and Solis-Oba, 2011). Per wave, the associated items are assumed to have an equal weight as compared with pick carts capacity. Therefore the cart capacity per wave is assumed fixed.

The problem in top of the hierarchy is the release moment of a batch. Docdata has agreed to process all orders placed in the web shop before a certain cut-off time, and therefore the moment of batching is restricted by the due date of an order. Postponement of the batch moment increases the number of available orders and thus enables the batch clustering to reduce the average travel distances per pick, and thus increases the pick performance (Pang and Muyldermans, 2013). The release moment decision is thus a trade-off between due date performance and pick performance. In case of Docdata all released orders have the same due date. A *time based frame* assures each demand is shipped within a predetermined time frame. Chen et al. (2005) found that the *quantity-based scheme* often outperforms its time-based counterpart, and no exception is found in which this is not the case. In order to optimize the pick run cutoff criteria, it should be investigated what the parameters of those schemes should be.

6. Research assignment

As mentioned in Chapter 5, the problem is decomposed into three sub problems. In line with these three problems, taking into account the scope of this project, the main research question is formulated as:

How can the batches be routed, clustered and released such that they minimize time devoted to picking?

This main research question will be answered by three research questions, with a number of associated sub-questions.

1. *What is the most efficient routing strategy, given a clustering strategy, to minimize time devoted to picking?*

In order to design an efficient routing strategy, the current strategy has to be mapped. In addition, as already mentioned, the routing strategy is strongly dependent of the number of picks per aisle. Therefore, the number of picks per aisle has to be mapped. Using this analysis, a routing heuristic should be chosen and compared with the current routing methodology. In order to answer research question 1, the following sub questions should be answered.

- a) *What is the current routing strategy?*
 - i. *How many picks per aisle are needed on average?*
- b) *What is the desired routing strategy?*
- c) *What is the difference in time between the current and the designed strategy?*

The second research question concerns how the batches should be composed such that the travel time can be minimized.

2. *How should orders be clustered, given a cut-off criteria, such that the time devoted to picking is minimized?*

In order to answer the second research question, insight should be acquired into the current clustering methodology, functional requirements for the design have to be composed and the batch size has to be calculated. With use of this information, a design for clustering should be made. Thereafter, this design should be compared to the existing batch clustering. Therefore the following sub questions are composed.

- a) *How are the orders currently clustered?*
 - i. *What are the critical aspects of order clustering?*
 1. *Under what conditions orders have to be batched separately?*
 2. *What are the functional requirements to cluster a batch?*
 - ii. *What is the current average time per pick?*
- b) *What is the anticipated clustering methodology?*
 - i. *What is the optimal batch size?*
 - ii. *What is the time per pick?*
- c) *What is the difference in time between the current and the designed methodology?*

The third question concerns at which criteria the batch should be released and how the cut-off value should be set:

3. *What should be the criteria to release a batch to minimize time devoted to picking?*

In order to answer this third research question, the current situation has to be mapped, a design should be made and potential benefits should be mapped answering the following research questions.

- a) *What are the current criteria to release a batch?*
 - i. *Which criteria affect the picking performance, and which parameters do not?*
- b) *How should the relevant criteria affect the release moment decision?*
 - i. *Should the release criteria be, quantity based, time based or time and quantity based?*
 - 1. *What should be the cut-off values for the quantity and/or time based criteria?*
- c) *What is the difference in time between the current and the designed methodology?*
 - i. *Does the release moment of batching result in monetary benefits?*

7. Design

The aim of this research is minimizing the time devoted to picking given the research assignment defined in Chapter 6 and the scope described in Paragraph 1.1.1 and 4.4. The design is made event based. The model is formulated by first defining the assumptions. Thereafter, the key performance indicators are defined in section 7.2. Then, the experimental setting is formulated in Paragraph 7.3. Finally, a scenario analysis is made in Paragraph 7.4.

7.1 Assumptions

In line with the process and problem description, some assumptions are made in order to simulate the picking process.

1. *Identical constant workforce*

In this research it is assumed that all pickers are identical in performance, and can pick every wave with the same efficiency during the day. This assumption is commonly made in literature (De Koster et al., 1999), (Roodbergen & De Koster, 2001), (Vaughan & Petersen, 1999).

Because it is costly to hire extra pickers during the day, a constant amount of pickers during the working hours are assumed. At Docdata pickers start at 8 AM on Monday and at 4 PM on all other days. Because the workforce on Monday between 8 AM and 4 PM is approximately half the workforce after 4 PM at all other days, it is assumed that pickers start at 12 PM noon on Monday and at 4 PM on all other week days. Furthermore, it is assumed that pickers receive their last batch at 12 AM midnight; they will always finish their batch.

2. *S-shape two-sided routing strategy*

At Docdata aisles can be accessed from both sides and can be travelled in both directions. Symmetric distances are assumed, which means that the distance from x to y is equal to the distance from y to x. This is realistic because of the rectangular shape of the aisles and racks. Within an aisle, two-sided order picking is performed. Because the aisle width is relatively narrow, the time to cross from the left side rack to the right side rack is negligible and assumed zero. For every wave, there is a fixed depot, located at corner of the warehouse from which every route starts and ends. For 99.45% of all waves executed, the number of picks per aisle exceeds 3.8 and thus the routing is assumed S-shape.

3. *Constant velocity, extraction time, floor switch time and start up time*

The travel time is assumed linear to travel distance and the travel speed of the pickers is constant and equal according to internal time count equal to 0,9 m/s. The constant velocity assumption is commonly made in literature on batching (De Koster et al., 1999), (Roodbergen & Vis, 2006), (Roodbergen & De Koster, 2010), (Nieuwenhuysse & De Koster, 2009).

It is assumed that the extraction time is independent of the composition of a batch. Because the requirement is that all items should be retrieved, the extraction times can be omitted from the batch clustering optimization problem altogether. To check whether all batches are handled in time, the extraction time is assumed constant irrespective of the outbound line, picker and type of item. Data analysis reveals that the average extraction time equals 7.7 seconds (Paragraph 4.6).

The time to switch a floor with use of the elevator is assumed constant, independent of the composition and size of a batch. Besides, the floor switch time is also independent on the combinations of floors because it is mainly composed of waiting in front of the elevator. Data analysis reveals that when one picker, picks one batch located at multiple floors, the time between two picks at separate floors is when eliminating outliers above 2 hours, on average 190 seconds. As explained in Paragraph 4.1, ten zones can be distinguished. While for certain zone combinations, no floor switch is required, every zone is defined as a separate floor, and every zone switch takes 190 seconds.

The time to start-up a batch includes pick up an empty pick cart and a printed batch list, this has to be done for all types of batches. Therefore, it is assumed that the startup time of a batch is independent on the wave type and the composition and size of the batch. In addition, the start-up time of a batch is assumed constant. Data analysis on the time between the scan of the batch form and the first pick reveals that the average start-up time is equal to 146 seconds.

4. *The capacity of the pick device per wave is constant*

It is assumed that all items allocated to a certain wave have an equal weight as compared to pick device capacity. The main objective to split up a wave is the fact that every outbound line can process only a limited amount of dimensions. Therefore the assumption that all items in a wave have the same dimensions is reasonable.

Furthermore, it is assumed that one single order will always fit in the pick device, and thus all orders can be retrieved completely. This assumption is realistic because the order sizes tend to be small in the e-commerce sector.

In addition, the batch sizes per wave are restricted by the number of sorting directions of the outbound line and the number of items per wave that fit in the pick device. The minimum of those two is given in the system. However, the historical batches are sometimes larger than this system capacity. In order to make fair comparison possible, it is assumed that the pick device capacity will never be overruled by the current batches and therefore the cart capacity per wave equals $\max(\text{Largest historical batch}; \text{system capacity})$.

5. *System stock equals actual stock*

Orders are only released by the order run, when the system displays there are enough items on stock. Therefore, it is assumed that the system stock is always equal to the actual stock.

6. *System calculation time can be neglected*

The time to create a batch is negligible. An IT manager at Docdata indicates that the time to create a batch will be approximately 1.5 second, which can be considered negligible.

7.2 Key performance indicators

In order to evaluate the performance of the defined scenarios, the key performance indicators (KPI's) should be defined. The objective of the process fulfilled at Docdata is that all orders are handled in time, and that the time devoted to picking is minimized. For the picking process this means that all orders should be handed over to the sorting and packing department before 12.30 AM. The due date performance is therefore defined as the percentage of batches which are finished after 12.30 AM. Aligned with the objective function of this research, the pick performance can be assessed by the total time needed per item. The performance of each scenario is thus twofold and based on both

pick performance and due date performance. The objective should be minimizing the time devoted to picking such that at most 0.2 per cent of the batches finish late.

1. Due date performance: Percentage of batches which finish after 12.30 AM.
2. Pick performance: Total time needed per item.

Because the defined KPI's determine the performance of the batch algorithm, these KPI's should be monitored during the simulation.

7.3 Experimental setting

This paragraph explains the experimental setting in which the design for the picking routing, batch clustering and release of batches is made.

7.3.1 Pick routing

When a batch is composed, the pick routing should find the sequence in which a batch should be routed. The pick routing is dependent on the numbers of picks per aisle. The number of picks per aisle is dependent on the put away strategy, the demand pattern and the overlap in retrieval area of the defined waves. The put away strategy and the demand are taken as given. In order to choose a pick routing, the minimum requirements to split up orders into waves should be determined. This is described in Appendix C, Paragraph 16.3.3. Given the minimum requirements for the waves, the numbers of picks per aisle are given in Table 4 Appendix A.

The pick routing can be determined by an optimal solution or a heuristic. Because of computational limitations in the current warehouse managing system, a heuristic is needed. Furthermore, the numbers of picks per aisle are relatively large which increases the relative performance of heuristics in comparison with the optimal solution, and therefore justifies the decision to use a heuristic for the pick routing (De Koster and Van Der Poort, 1998).

Whether the largest gap or S-shape strategy is the most appropriate depends on the number of picks per aisle (Hall, 1993). Data analysis reveals that for 99.45% of all waves executed, the numbers of picks per aisle per day exceed 3.8 and the S-shape strategy performs better than the largest gap strategy. Therefore the S-shape strategy is chosen as routing strategy.

7.3.2 Batch clustering

Given that the S-shape routing strategy will be applied and the fact that the batching problem is NP-hard, the defined waves should be clustered according to the seed algorithm (De Koster et al., 1999). The seed algorithm is shown in Figure 15. The objective of batch clustering is to minimize the time devoted to picking and the objective function is given in Formula (6) Paragraph 5. De Koster et al. (1999) have already shown that the cumulative seed rule hardly performs better than the single seed rule. Because the cumulative seed rule is more complex than the single seed rule, the single seed rule is recommended.

The question remaining is what type of seed selection and order addition should be chosen. It is assumed that one single order always fits in a pick device. Thus the capacity of a pick device will not be taken into consideration during the seed selection. The seed selection rule can be determined

internally or externally. When it is determined internally, an initial order is chosen which is available in the system. When the seed selection rule is determined externally, an area is determined in which the orders can be placed. Whether it is better to determine the seed internally or externally depends on the uniqueness of the aisle and floor combinations in an order. As shown in Figure 13, the aisle and floor combinations of the orders are rather similar and thus an internal seed selection methodology is preferred. The type of seed selection and order addition rules are examined using a scenario analysis which is described in Section 7.4.

7.3.3 Release of batches

The aim of the release policy at Docdata is ensuring that all orders are handled in time. A crucial link between order picking and due date performance is that the faster an order can be retrieved, the sooner it is available for the next process (De Koster et al., 2006). In order to keep the order pickers operating, every time an order picker returns at the depot, a batch should be available. More efficient batches can be made when more information is available. Because a constant workforce is assumed and the time to create a batch is assumed negligible, batches should only be created when a picker is available.

As indicated in Paragraph 4.3, all orders placed before 11 PM have to be picked before 12.30 AM, which is only one and a half hour. In addition, the numbers of requested orders are expected to increase. Given the increased pressure on short delivery lead times, minimizing order throughput times is an important objective (van Nieuwenhuysse & de Koster, 2009). Van Nieuwenhuysse and De Koster (2009) distinguish two effects which influence the throughput: the batching effect and the saturation effect, both are explained sequentially. When the numbers of orders per batch are small, the extraction time and travel time are small, but the waiting time will be large. This is due to the fact that with small values the numbers of orders setups need to be performed frequently, which causes the utilization of the pickers to increase. Consequently, the batches will need to wait longer in queue before a picker becomes available, which increases the throughput time of the orders. This effect is called the *saturation effect*. In contrast, when the numbers of orders are large, the waiting time will be small because fewer setups need to be performed. On the other hand, the extraction time and travel time will be large, and will be the dominant aspects of the throughput time. This effect is called the *batching effect*. The combination of the saturation and batching effect yields in a convex relationship, implying the existence of an optimal number of orders per batch (van Nieuwenhuysse & De Koster, 2009). There are multiple waves defined at Docdata and it is assumed that every order picker can pick every wave with the same efficiency. When more orders are available before batch clustering, the opportunity to create more efficient combinations of orders increases. In addition, saturation should be prevented, because when an order picker starts a tour for a wave with a limited amount of orders, the workforce capacity may be insufficient to process all orders (Le-Duc & De Koster, 2007). Therefore the wave with the most picks available compared to the associated cart capacity should be released first.

When a picker is available and the wave is chosen, the decision should be made whether this wave should be released or not. Pang and Muyldermans (2013) found that postponement of the releasing decision can result in a significant reduction in routing distance especially when the capacity of the pick device is relatively large and the demand is not bulky. Chen et al. (2005) found that quantity

based releasing policies outperform time based releasing policies, and therefore a quantity based releasing policy is preferred. The cut-off criterion which indicates the pick efficiency should be defined in the scenario analysis. In order to prevent saturation and utilize the benefits of postponement, when too few orders are available, the order picker has to wait five minutes and then the calculation is repeated. In addition, a time based releasing policy is needed to ensure that all orders are retrieved in time. After 11 PM, no additional orders will come available. Therefore a time based releasing policy with the cut-off value 11 PM, should ensure that all orders are released in time. The release methodology is given in Figure 15.

7.4 Scenario analysis

This paragraph discusses the degrees of freedom in the design and the scenarios that will be simulated, in order to answer the second and third research question.

7.4.1 Degrees of freedom

The experimental setting is shown in a flow chart in Figure 15. Given the experimental setting, there are three degrees of freedom for the scenario analysis; those are marked yellow in Figure 15. Namely, how the waves are selected, what the seed selection rule is and how orders are added to this seed. As the flow chart shows, the orders pass the model in one direction. Therefore the model is built up event driven, and three events can be distinguished. In the first state (state 0), the order is not yet placed in the web shop, and the order is thus unknown by Docdata. In the second state (state 1), the order becomes available in the system. Thereafter, in the third state (state 2), a batch is created for this order and picked respectively. Those events are monitored by the simulation.

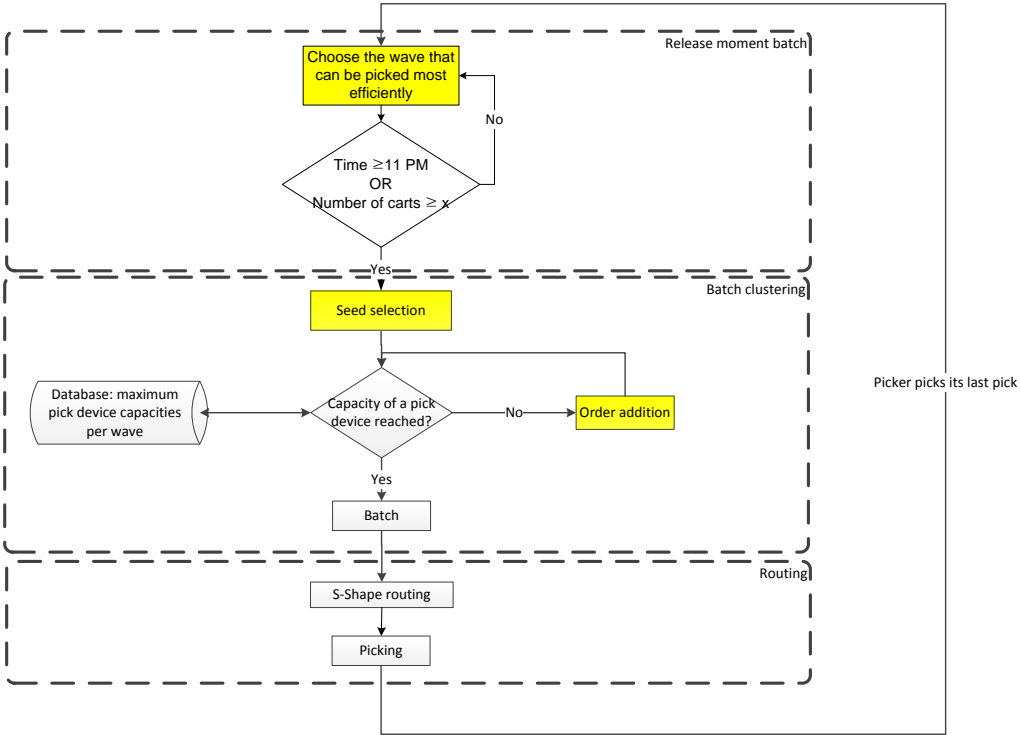


Figure 15 Experimental setting

7.4.1.1 Wave selection criterion

The *wave selection criterion*, determines in which order the waves, w , (as given in Appendix C, Paragraph 16.3.3) are released. As given in the experimental setting a time and quantity based releasing policy is needed. The quantity based releasing policy should release those waves which can be picked most efficiently. A batch becomes more efficient when it contains a sufficient number of orders to choose from. Therefore the number of items per wave w , should be maximized. In order to assess whether a wave is efficient, the number of filled pick carts per wave y at time t , $FPC_{Y,t}$, should be calculated. This can be done by the following formula:

$$FPC_{Y,t} = \frac{\text{Items available wave } Y \text{ at time } t}{\text{Pick cart capacity wave } Y} \quad (10)$$

When the number of filled pick carts per wave y at time t , $FPC_{Y,t}$, is low, the startup time weights more heavily and the *saturation effect* occurs (van Nieuwenhuysse & de Koster, 2009). The saturation effect causes an increased throughput time of orders, which is unfavorable both in terms of pick and due date performance. Therefore, a wave should only be released when the number of filled pick carts per wave is higher than a predetermined amount. This cut-off value of the quantity based releasing policy cannot be too high either, because this increases the pressure on the picking department which probably results in a decrease of due date performance. The number of filled pick carts per wave y should be determined by a trial simulation run, see Paragraph 8.2.

When the maximum number of filled pick carts over all waves at time t is equal for more than one wave, outbound should be facilitated as much as possible. The objective of the outbound lines is preventing starvation. Starvation of the outbound lines can be calculated by the run out. The run out is dependent upon the hourly sorting capacity per outbound line and the supply of items to that line. Because the outbound process is out of scope, the sorting capacities are assumed deterministic. The sorting process will process all orders every day and the picking department starts the day one hour before the sorting department. Therefore, the supply of items to the outbound line is only based on the number of items which finished the picking process the last hour. The run out of the outbound line x at time t , $RO_{x,t}$, can therefore be calculated by the following formula:

$$RO_{x,t} = \frac{\text{Number of items sent to outbound line}_x \text{ between } [t - 3600 \text{ seconds}, t]}{\text{Hourly sorting capacity outbound line}_x} \quad (11)$$

When $RO_{x,t}$ decreases, the need for batches at outbound line x increases. The following methodology should thus be applied.

1. When a picker is available, calculate $FPC_{Y,t}$ for all waves. Order them in decreasing sequence. When the maximum $FPC_{Y,t}$ is equal for more than one wave, choose the wave with the largest run out, $RO_{x,t}$. Choose the first wave in this sequence.
2. Determine for the chosen wave whether $FPC_{Y,t}$ exceeds the predetermined amount or the time exceeds 23.00 hours. When this is the case, release the wave. Otherwise wait 300 seconds and start at step 1.

7.4.1.2 Seed selection

The objective of the seed selection is: minimizing the total time devoted to picking. Wolters (1996) shows that 'largest, longest or farthest' seed selection rules constantly outperform the 'smallest, shortest and nearest' rules in a single floor warehouse. The time per order g is composed of extraction time of the items, travelling time and floor switch time $(e_t i_g + \frac{d_g}{v} + t_v n_{v,g})$. When the time per order is large, it is harder to add this order to a route later on. Therefore the seed rule *longest processing time* (LPT) is chosen. When the maximum processing time is equal for more than one order, choose the order with the maximum number of items, because these orders are harder to add later on. Thus, the seed selection rule LPT follows the following steps:

1. Order all orders g in decreasing sequence of processing time $(e_t \cdot i_g + \frac{d_g}{v} + t_f \cdot f_g)$. When the maximum processing time is equal for more orders, choose the order with the maximum items in an order.
2. Choose the first order in the sequence as seed order.

In the current situation orders go over far more floors than what is strictly needed (Figure 8). Because the time required to change a floor is high, the number of floors should be minimized. To prevent that a large amount of batches will go over multiple floors, another seed selection rule is designed. In this seed selection rule, the orders should be ordered on the number of floors required in increasing order. For the orders that should traverse the minimum number of floors, the order with the longest processing time (LPT) is chosen. This rule is defined in this research as floor longest processing time (FLPT), and the methodology is described below.

1. Order the orders on the number of floors, f_g , in increasing sequence.
2. Order the orders with the minimum number of floors needed ($\min(f_g)$), in non-increasing sequence based on the processing time $(e_t \cdot i_g + \frac{d_g}{v} + t_f \cdot f_g)$. Take the first order in that sequence.

7.4.1.3 Order addition

The aim of the order addition rule is minimizing the time needed for picking. The time needed for picking is fourfold and composed of travel time, elevator switch time, startup time and extraction time. The order addition should thus minimize the sum of those times. In order to acquire processing time minimization, the savings is acquired by adding an order j to the seed order i should be computed. The savings, $S_{i,j}$, can be computed by extracting the time of retrieving order i and order j simultaneously from the time of retrieving order i and order j individually.

$$Time_i = e_t \cdot i_i + \frac{d_i}{v} + t_f \cdot f_i + st \quad (12)$$

$$Time_j = e_t \cdot i_j + \frac{d_j}{v} + t_f \cdot f_j + st \quad (13)$$

$$Time_{i,j} = e_t \cdot i_{i,j} + \frac{d_{i,j}}{v} + t_f \cdot f_{i,j} + st \quad (14)$$

$$S_{i,j} = \frac{d_i + d_j - d_{i,j}}{v} + t_f (f_i + f_j - f_{i,j}) + st \quad (15)$$

Because the saving should be maximized, the saving should be ordered in decreasing sequence. Orders can only be added to the seed when the saving is positive. When the maximum saving is equal for more orders, choose the order with the maximum number of items, since this order is harder to add later on. This methodology can be described as follows.

1. For all not assigned orders n compute the savings of adding order j to seed order i

$$S_{i,j} = \frac{d_i + d_j - d_{ij}}{v} + tf(f_i + f_j - f_{i,j}) + st$$
2. Determine whether the cart capacity of the associated wave is filled. When this is the case, stop, otherwise go to step 3.
3. Order the savings $S_{i,j}$ in decreasing sequence. When the maximum saving is equal for more orders, choose the order with the maximum number of items. If the savings $S_{i,j}$ are positive (>0), add the first one in the row to the existing route. Go to step 1.

The assumption in this research is that the startup time of a new batch is less than the floor switch time ($t_v > st$). Based on this assumption, computational complexity can be reduced by not permitting an order to add an extra floor to the seed. Therefore, when a seed order contains x floors, the orders that are added to this seed, should not add other floors to the route. Therefore the following methodology should be followed.

1. Determine all orders n which contain no additional floors compared to seed order i . Thus all orders j for which $f_{i,j} = f_i$ holds.
2. For all not assigned orders n compute the savings of adding order j to seed order i

$$S_{i,j} = \frac{d_i + d_j - d_{ij}}{v} + st$$
3. Determine whether the cart capacity of the associated wave is reached. When this is the case, stop, otherwise go to step 4.
4. Order the savings $S_{i,j}$ in decreasing sequence. When the maximum saving is equal for more orders, choose the order with the maximum number of items. If the savings $S_{i,j}$ are positive (>0), add the first one in the row to the existing route. Go to step 1.

It is common in literature to cluster the orders either according a seed algorithm or based on savings criteria. In literature however a limited set of orders are considered. De Koster (1999) for instance takes a set of 30 orders, while this research considers on average 22253 orders per day. For computational complexity reasons, the seed order is chosen based on processing time and not based on savings. When the seed order was chosen based on savings, the savings of all combinations of orders should have to be determined, while using this methodology only the savings compared to the seed order should be determined.

In addition, the processing time of an order gives a relatively good approximation of the savings. The processing time of an order is namely dependent upon the number of items in an order, the number of aisles, the number of cross aisles and the number of floors. When the processing time is large, the possibility to acquire large savings increases.

For comparison, the order addition in which the orders which take the longest processing time (LPT) are added to the seed is added.

7.4.2 Definition of scenarios

By combining the different settings for the defined degrees of freedom, six scenarios are defined. For a benchmark, the current Docdata algorithm is added as a reference. The scenarios are defined in Table 1.

Table 1 Definition of scenarios

Scenario	Wave selection	Seed selection	Order addition
Current	Manually	Order with the lowest location number of the first order line	Order with the lowest location number of the first order line
AAA	Largest $FPC_{Y,t}$ (tie smallest $RO_{x,t}$)	LPT	Savings
AAB	Largest $FPC_{Y,t}$ (tie smallest $RO_{x,t}$)	LPT	Floor savings
AAC	Largest $FPC_{Y,t}$ (tie smallest $RO_{x,t}$)	LPT	LPT
ABA	Largest $FPC_{Y,t}$ (tie smallest $RO_{x,t}$)	FLPT	Savings
ABB	Largest $FPC_{Y,t}$ (tie smallest $RO_{x,t}$)	FLPT	Floor savings
ABC	Largest $FPC_{Y,t}$ (tie smallest $RO_{x,t}$)	FLPT	LPT

8. Simulation model

Simulation is used to evaluate and compare the scenarios defined in Paragraph 7.4. The simulation model is written in Visual Basic for Applications (VBA) and the simulation runs are executed in Microsoft Access. The simulation model makes use of events, in which orders are processed in one direction. The behavior of those orders is monitored with use of those events. In addition, it is assumed that when a picker starts a batch he will always finish the batch, no preemption is allowed. The model is thus picker driven. Furthermore, the model is terminating, for all the defined scenarios (except the current scenario); no batches are released after 12 midnight.

This chapter encompasses a description of the data used, the validation and verification of the calculations in the simulation and the parameter settings for simulation.

8.1 Description of the used data

The amount of data for simulation should be large enough to take adequate conclusions. However, the amount should not be too extensive because this increases the calculation time enormously. Considering these aspects, a representative period of one week is chosen. This means that no machine breakdowns should occur. Roughly, two types of weeks can be distinguished at Docdata: a peak week and a non-peak week. Because the major part of the year consists of non-peak weeks, the simulation will be done on a non-peak week. Therefore in consultation with logistic engineers at Docdata the period from 10.03.2014 to 14.03.2014 is chosen as representative data set. With use of this data set, the time to run a single scenario takes on average 40 minutes (on a notebook with an Intel® Core2Duo T9600 processor). To be able to analyze the seven different scenarios and to repeat scenarios, this run length is reasonable. In addition, during this period batches are created for 111,266 orders, which is a larger quantity than used in common literature (De Koster, 1999). Due to the larger amount of orders, more reliable conclusions can be drawn. The data is explained in Appendix C Paragraph 16.3.4.

8.2 Validation and verification of the calculations

Validity is checked in several ways both during the model construction and afterwards, individual modules are checked manually to verify whether they are doing what was intended. For instance, the distance for the batches is calculated using a manual calculation in Microsoft Excel and calculated by the simulation. When performing a regression with those two calculations, an adjusted r^2 93.3% is acquired, which is good. 100% adjusted r^2 cannot be achieved because the time between two sections is assumed 190 seconds in the simulation while a floor switch is not always required. In the manual calculation, the times for hall switches are determined by data analysis.

8.3 Parameter settings

The performance of the scenarios can only be measured under specific parameter settings and is strongly dependent upon the values of the parameters. Given the research design, the parameters that should be set are:

- The minimum number of filled pick carts per wave y at time t ($FPC_{Y,t}$) at which wave y can be released;
- The size of the workforce.

The parameter decision is based on the key performance indicators and is two dimensional. The key performance indicators are defined in Paragraph 7.2. The aim of the parameter setting is decreasing pick performance (total time needed per item), such that maximum 0.2 per cent due date performance (percentage of batches which finish after 12.30 AM) can be achieved. The effect of the parameters on the performance should be revealed, this is done by testing hypotheses. Those hypotheses are tested by a trial simulation run.

First, the minimum number of filled pick carts at which a wave can be released is determined. When the minimum number of filled pick carts increases, the time before a batch is released increases. This results in a more tight due date, thus it is expected more batches finish late (in other words: the due date performance increases). Therefore, the following hypothesis is posed.

Hypothesis 1: There exist a positive relation between the minimum number of filled pick carts per wave at which a wave can be released and the due date performance.

This relationship between the minimum number of filled pick carts per at which wave y can be released and due date performance is shown graphically in Figure 16. As can be seen the due date performance increases when the minimum number of filled pick devices increases. Linear correlation showed that a significant linear positive relationship exists, which confirms Hypothesis 1. The due date performance is allowed to be maximum 0.2 per cent, therefore the maximum number of filled pick carts per wave at which a wave can be released is equal to 0.8.

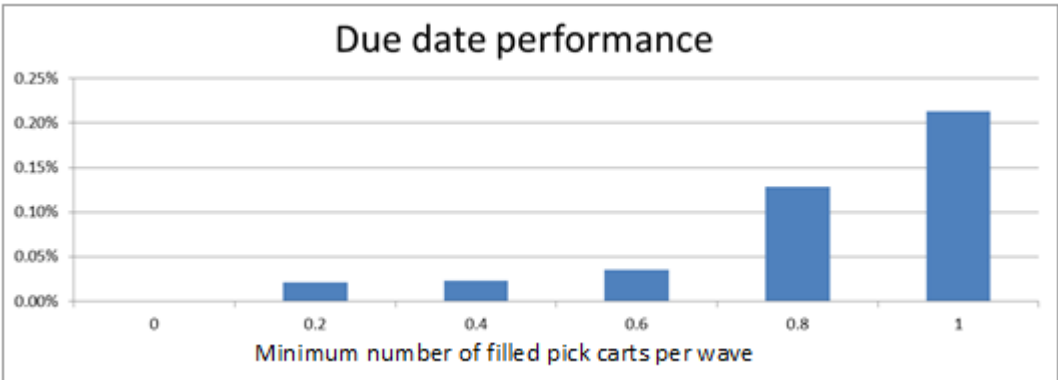


Figure 16 Relationship number of filled pick carts per wave y at time t ($FPC_{y,t}$) at which wave y can be released and due date performance

When the minimum number of filled pick carts before release is far below 1, the numbers of orders per batch are small. This results in small extraction time and travel time, but the waiting time will be large. This is due to the fact that, with small values the numbers of orders, setups need to be performed frequently which causes the utilization of the pickers to increase. Consequently, the batches will need to wait longer in queue before a picker becomes available, which increases the throughput time of the orders (Van Nieuwenhuysse & De Koster, 2009). When this saturation effect occurs, the pick performance will increase.

Hypothesis 2: A negative relation exists between the minimum number of filled pick carts per wave y at time t ($FPC_{y,t}$) at which wave y can be released and the pick performance.

The relationship between pick performance and minimum number of filled pick carts per at which a wave can be released is given graphically in Figure 17. In line with the expectation, the figure shows that the pick performance decreases when the number of filled pick carts increases. Linear correlation also showed a negative linear correlation between the pick performance and the number of filled pick carts before release, and thus Hypothesis 2 is confirmed. In order to increase pick performance, the minimum number of filled pick carts per wave y at time t ($FPC_{Y,t}$) at which wave y can be released should be maximized, given the restriction by due date performance, 0.8 filled pick carts per wave should be available before a wave is released.

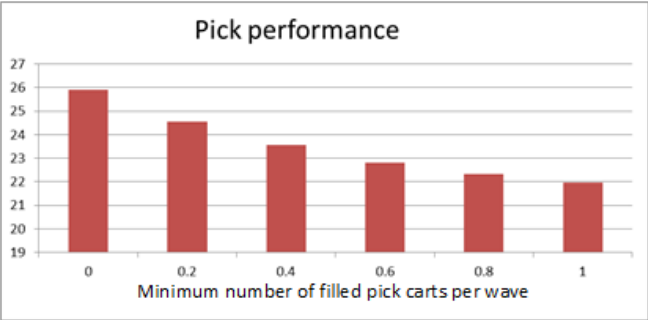


Figure 17 Relation number of filled pick carts per wave y at time t ($FPC_{Y,t}$) at which wave y can be released and pick performance

Given that the preferred number of filled pick carts per wave y at time t ($FPC_{Y,t}$) at which wave y can be released is 0.8, the parameter value for the workforce is determined. When the number of pickers, P , decreases, the pressure on those pickers increases, which is expected to result in more late finished batches.

Hypothesis 3: The workforce is negatively related to the due date performance.

For all scenarios, the due date performance is calculated, for different workforces ($P = 35,40,45$). The relation between the due date performance and workforce is given graphically in Figure 18. As the figure shows the due date performance (percentage of late batches) decreases when the number of pickers available increases. In line with the expectation, linear correlation shows a significant ($p=0.05$) negative correlation between the workforce and the due date performance, which confirms Hypothesis 3. At most 0.2 per cent of all batches are allowed to finish late, thus at least 40 pickers should be chosen.

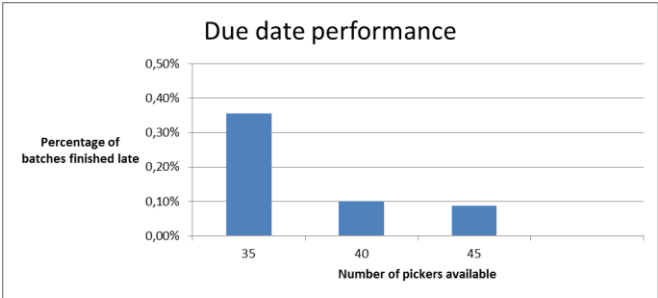


Figure 18 Relationship between workforce and due date performance

On the other hand, when decreasing the workforce, orders have to wait longer before a picker is available and thus before they are clustered into a batch. Postponement of the batch moment increases the number of available orders and thus enables the batch clustering to reduce the average travel distances per pick, and is expected to result in an increased pick performance (Pang and Muyldermans, 2013).

Hypothesis 4: There exists a positive relation between the pick performance and the workforce.

For every scenario, the time per item is increasing when the workforce increases. The relationship is given in Figure 19. The figure shows that the pick performance increases when the number of pickers available increases. In contrast to the expectation, linear correlation has shown that the relationship is not significant ($p=0.05$), and therefore the hypothesis is not accepted. This can be caused by the fact that ineffective batches are hold until the number of filled pick carts per wave y at time t ($FPC_{y,t}$) at which wave y can be released of 80 per cent is reached. Because the pick performance is better for 40 than for 45 pickers, the parameter setting of 40 pickers is chosen.



Figure 19 Relationship between workforce and pick performance

9. Results

Given the parameter settings explained in Paragraph 8.2, the results for the different scenarios defined in Paragraph 7.4.2 are explained in this chapter.

9.1 Batch clustering

In this paragraph, an overview of the pick and due date performance of the defined scenarios is given. A deeper explanation will be given in subparagraphs. The explanation of the scenarios is given in paragraph 7.4.2. The pick performance (Total time needed per item) for the different scenarios is shown in Figure 20, and indicates that scenario AAA and AAB outperform all other scenarios based on pick performance. The due date performance (Percentage of batches which finish after 12.30 AM) is given in Figure 21. As given in Figure 21, simulation has showed that scenario AAB, ABA (and ABB) outperform all scenarios based on due date performance.

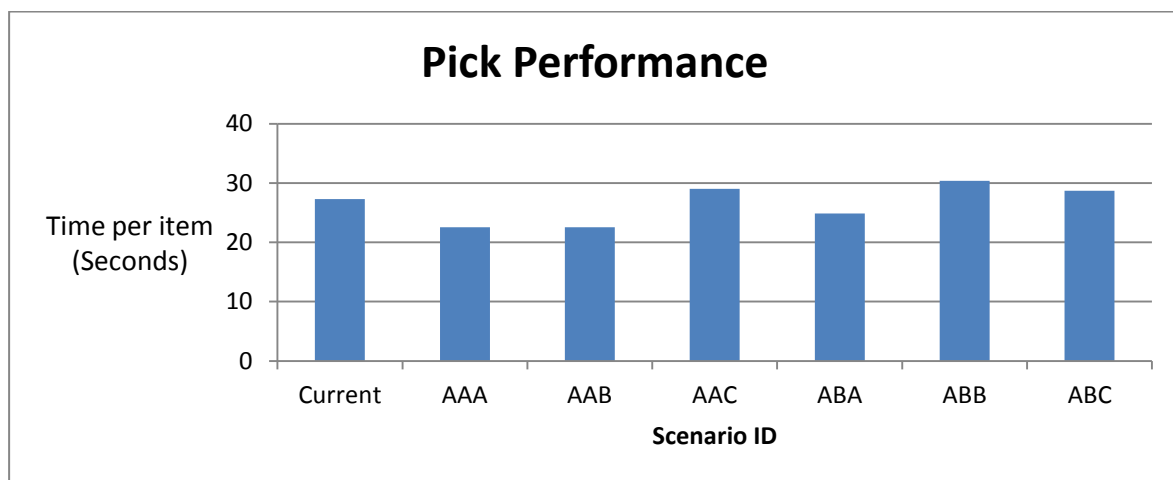


Figure 20 Pick performance for the defined scenarios

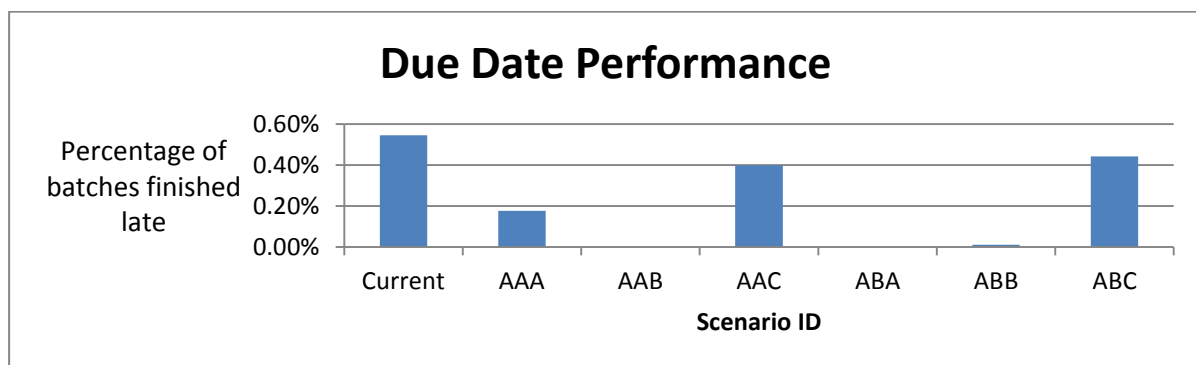


Figure 21 Due date performance for the defined scenarios

To understand the performance of the defined scenarios, the effect of the seed selection and order addition rules should be evaluated. This analysis is performed in Paragraph 9.1.1 and 9.1.2. This analysis is performed drawing boxplots, in which the minimum, maximum, median and two quartiles are shown.

9.1.1 Seed selection

Given the parameter setting, the performance of the seed selection is evaluated on the key performance indicators: pick performance and due date performance. Two degrees of freedom are defined namely longest processing time (LPT) and floor longest processing time (FLPT) seed selection. The performance of the seed selection rules are given in Figure 22 and Figure 23. As can be seen in the figures, the LPT seed selection rule always outperforms the FLPT seed selection rule in terms of pick performance. This can be explained by the fact that the FLPT seed selection will barely never choose a seed order which contains many floors, while those orders are harder to add later to the batch. Because, it is harder to add an order which is located at many floors to a route later on, this will result in less efficient batches and thus results in extra picking time.

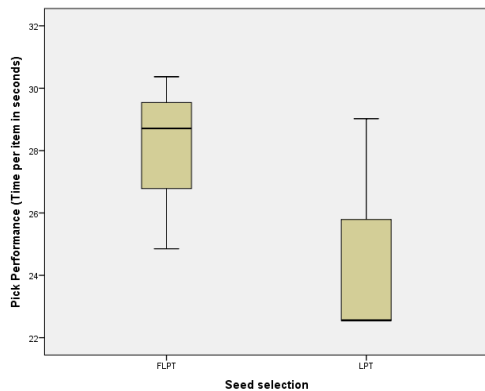


Figure 22 Pick performance on seed selection

Besides to the performance of the seed selection rule in terms of pick performance, the due date performance should be evaluated as well. The due date performance is given in Figure 23. As can be seen, the FLPT seed selection performs slightly better than the LPT seed selection in terms of due date performance. However, in the LPT seed selection still 99.9 per cent of all batches are finished in time.

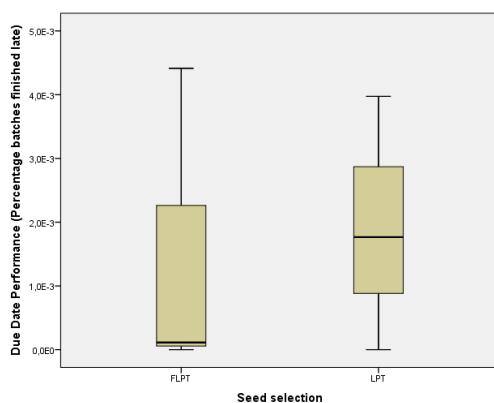


Figure 23 KPI Due date performance on seed selection

Because contradictory performances are found, a trade-off should be made between due date performance and pick performance. Because both seed selection rules perform better on the due

date performance than the minimum required value of 0.2 per cent, the decision is made on the pick performance. Therefore the LPT seed selection rule is chosen.

9.1.2 Order addition

Given the parameter setting, the performance of the order addition rules will be evaluated both on pick and due date performance. For the order addition, there are three degrees of freedom: floor savings-, savings- and LPT order addition. The performance of the three order addition rules are given in Figure 24 and Figure 25. As can be seen in Figure 24, the worst pick performance of the savings order addition outperforms the best pick performance of the LPT order addition. This can be explained by the fact that the LPT order addition does not take into account the location of the seed order which leads to inefficient combinations of orders. The other order addition methodologies aim to minimize the additional time when adding orders to the seed.

Furthermore, the savings order addition outperforms the floor savings order addition in terms of average pick performance. While the best pick performance of the floor savings order addition exceeds the best pick performance of the savings order addition. The large variance of pick performance of the floor savings order addition compared to the savings order addition can be explained by the fact that the savings rule assesses whether an extra floor switch can result in a saving, while the floor savings rule never allows an extra floor switch. In the savings rule a trade-off between startup time, floor switch time and distance is made, while in the floor savings order addition only a trade-off between startup time and travel time is made. In other words, in the floor savings rule, a degree of freedom is taken away, which is only expected to lead to good pick performance given a limited amount of parameter settings. Under what specific conditions, which order addition is preferred, should be evaluated by a sensitivity analysis.

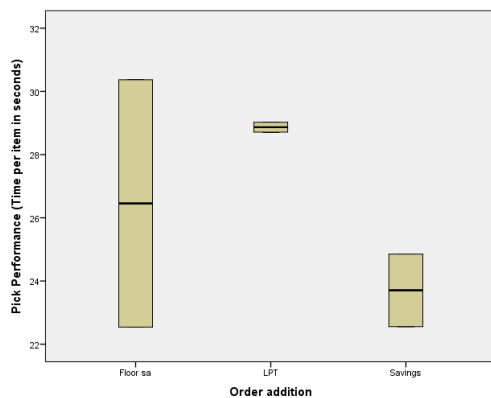


Figure 24 Pick performance on order addition

The due date performance is given in Figure 25. The LPT order addition rule performs the worst; both in terms of pick performance and due date performance and therefore the LPT order addition rule will never be preferred. Because the LPT order addition rule, does not take into account the location of the seed order an one-dimensional decision based on processing time is made. By not taking into consideration some information, less efficient combinations of orders are made, which increases the duration of the picking routes. Due to the longer duration, there is more pressure on the picking department and therefore the due performance increases. The floor savings order addition, performs

better in terms of due date performance than the savings order addition. However, this is only a minor difference. Therefore a trade-off between floor savings and savings order addition should be made in the sensitivity analysis.



Figure 25 KPI Due date performance on order addition

9.2 Comparison with current situation

As elaborated, the LPT seed selection and the (floor) savings order addition are preferred. Therefore, scenario AAA or scenario AAB are preferred and should be compared with the current clustering methodology. In terms of pick performance, 17.4 per cent savings in time can be acquired by applying the savings order addition, and 17.5 per cent savings in time can be acquired by applying the floor savings order addition instead of the current methodology. The pick performance is three dimensional and based on travel time, floor switch time and startup time. The percentage change per element is given in Figure 26. Scenario AAA, which uses the savings order addition, performs better on the travel time and startup time. This can be explained by the fact that the savings order addition takes more degrees of freedom into consideration. Scenario AAB, which uses the floor savings order addition, performs better on the floor switch time. This is as expected, because this order addition rule does not allow extra floor switches. Consequently, scenario AAB results in an increase in startup time compared to the current scenario. The floor savings rule namely does not make the consideration whether switching a floor results in less time than starting up a new batch, and will always start up a new batch.

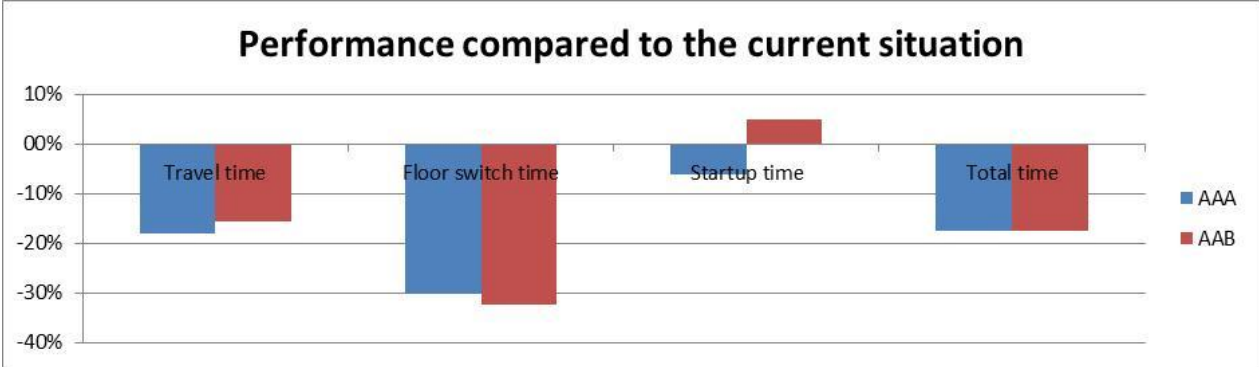


Figure 26 Percentage change compared to the current scenario

Additional to the pick performance, the due date performance should be compared with the current methodology. The due date performance of the current methodology improves by either applying scenario AAA or scenario AAB. Thus by applying either scenario AAA or AAB both pick and due date performance can be improved.

9.2.1 Practical usefulness

The practical usefulness of this project lies in the opportunity to implement a new batch clustering methodology. By selecting a seed order based on LPT and adding an order based on (floor) savings, savings in both pick and due date performance can be acquired. By discussing the methodology with both the IT managers and the operations managers at Docdata in early stages of the project, acceptance is created. By extensively explaining the idea, those managers are convinced and the implementation of this project has already started. The IT department is building a program which handles the batching logic. When this trial program achieves the desired performance, the program will be extended to the real data of one client. When the new clustering methodology achieves desired performance in an industrial setting, the methodology can be rolled out to other customers as well.

10. Sensitivity analysis

A sensitivity analysis is performed to test the sensitivity of the model to certain parameters. This sensitivity analysis is performed to reveal when the savings order addition and when the floor savings order addition is preferred. Moreover, the sensitivity analysis is used to reveal the effect of multiple floors on pick performance. Afterwards, the effect of the made assumptions is revealed.

10.1 Trade-off between savings and floor savings order addition

The major difference between the savings- (Scenario AAA) and the floor savings (Scenario AAB) order addition is that the savings order addition evaluates whether an extra floor switch can result in a saving, while the floor savings order addition never allows an extra floor switch. In order to achieve the above mentioned, the savings order addition makes a trade-off between startup time, floor switch time and travel time, while the floor savings order addition only makes a trade-off between startup time and travel time. In other words, in the floor savings order addition, a degree of freedom is taken away, which is only expected to lead to good pick performance given a limited amount of parameter settings. Because no clear preference between the savings- and the floor savings order addition can be pinpointed, and the only difference is on the evaluation of the floor switch time, the sensitivity of the pick performance to the floor switch costs have to be evaluated.

For the sensitivity analysis the floor switch time is varied between 0 and 300 seconds with steps of 50 seconds. The pick performance for the two scenarios is given in Figure 27.

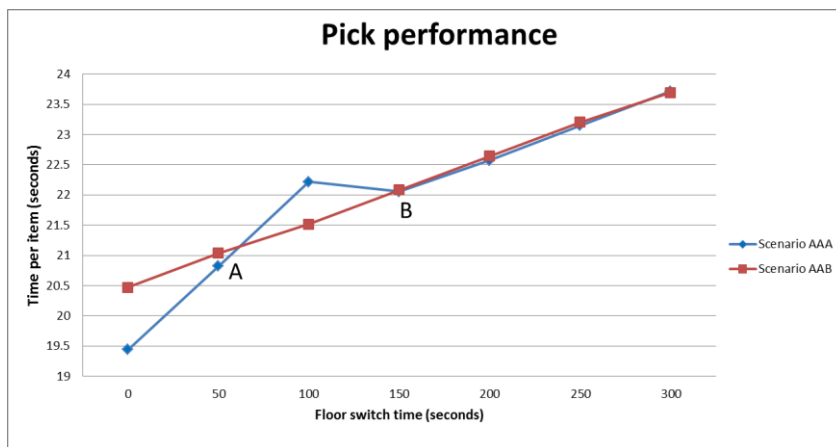


Figure 27 Trade-off between floors savings order addition and savings order addition

As Figure 27 indicates, three intervals of pick performance can be distinguished: low ($t_v < A$), high ($t_v > B$) and medium floor switch time ($A < t_v < B$). When the floor switch time is low ($t_v < A$), it takes less time to do an extra floor switch compared to starting up a new batch. The savings order addition makes this consideration and allows an extra floor switch during the order addition. The floor savings order addition never allows an extra floor switch. Because allowing a floor switch is beneficial given this floor switch costs, the pick performance of the savings order addition exceeds the pick performance of the floor savings order addition.

When the floor switch time is high ($t_v > B$), the savings rule will make a trade-off between startup time, floor switch time and travel time and will result that starting up a new batch is more beneficial

than adding a floor to the route. The floor savings rule, does not allow an extra floor switch during the order addition either and thus both the savings and floor savings order addition will result in the same pick performance.

In the intermediate case, when the floor switch time is medium ($A < t_v < B$), the floor savings order addition slightly outperforms the savings order addition in terms of pick performance. Although, the savings order addition takes into consideration more degrees of freedom, the decision is made in single instead of cumulative mode. Given that the savings order addition is applied in single mode, each iteration determines which not yet clustered order results in the largest saving compared to the seed order. Each iteration can thus add an order which is located at another floor compared to the existing route, while actually time savings can be achieved by adding orders which are located at the same floors as the previous added orders x . This is inefficient, and because the floor savings order addition will never allow an extra floor, the floor savings order addition is preferred in this interval.

Based on this sensitivity analysis, the following can be stated for the preference of floor savings or savings order addition:

1.
$$\begin{cases} Pick\ performance_{Savings} > Pick\ performance_{Floor\ savings} & \text{when } t_v < A \\ Pick\ performance_{Savings} \leq Pick\ performance_{Floor\ savings} & \text{otherwise} \end{cases}$$
2.
$$Due\ date\ performance_{Savings} > Due\ date\ performance_{Floor\ savings}.$$

Because the current floor switch time is 190 seconds and thus exceeds A, the floor savings order addition is preferred both based on pick and due date performance. In addition, fewer combinations should be evaluated using the floor savings order addition, which reduces the computational complexity.

10.2 Effect of multiple floors

The main contribution of this research is designing a methodology to cluster orders when a warehouse consists of multiple floors. The exact effect of multiple floors on the pick performance will be evaluated in this paragraph. Therefore two types of warehouse situations are defined. The first warehouse is the warehouse where the ten zones are placed above each other, conceptually as given in Figure 28.

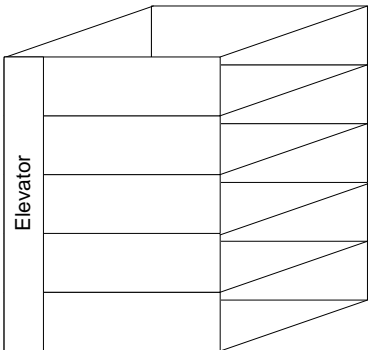


Figure 28 Warehouse one

The second warehouse has instead of multiple floors one floor. This means that the zones can be placed in succession, next to each other or a combination of both. When the zones are placed next to each other the aisle length increases. When there are fewer but longer aisles, the probability to skip an aisle decrease, which will increase the horizontal distance (Roodbergen & Vis, 2006). Therefore, placing the aisles in succession is preferred. All other characteristics, such as aisle length, cross-aisle length, number of aisles, are the same for both warehouse situations. When the zones are placed in succession, the sequence in which the zones are placed after one other should be determined. Because the total cross aisle length of every zone is different, the number of items required per cross aisle meter should be determined for all zones. The zone with the most picks per meter cross aisle, should be placed the closest to the depot. This is given conceptually in Figure 29. This figure shows besides the sequence of the zones, the number of picks per meter cross aisle between brackets and the time to the first aisle of the zone.

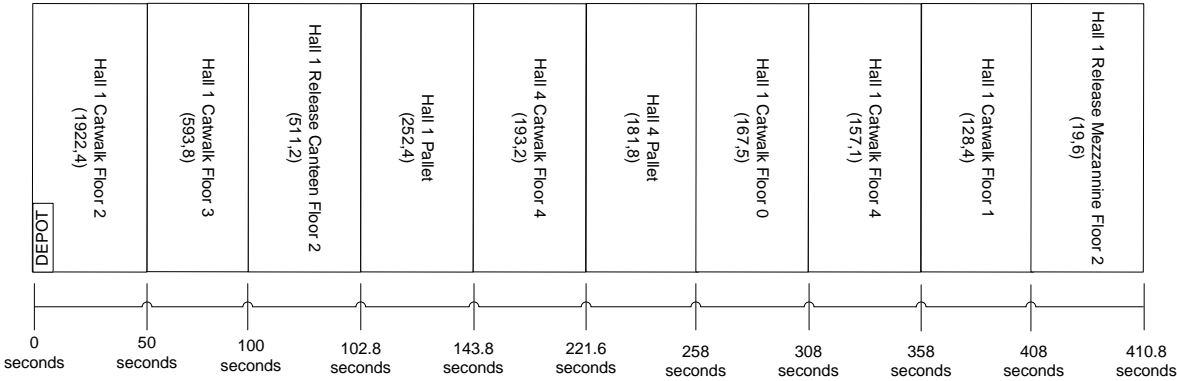


Figure 29 Warehouse two

The different types of warehouses are compared on pick performance. For warehouse one, the floor switch time is varied between 0 seconds and 300 seconds with steps of 50 seconds. For warehouse one, it is shown in Paragraph 10.1 that scenario AAA and scenario AAB outperform each other for different values of floor switch time, tf . For each floor switch time (tf), the best performing scenario is shown in Figure 30. The inflection point at a floor switch time of 50 seconds can be explained by the minimum of the scenarios AAA and AAB.

Then the pick performance for warehouse two should be determined. The only differences in time between the two warehouse situations are the floor-switch time and the cross-aisle time. The floor-switch time for warehouse two is equal to zero ($tf \cdot f_s = 0$), because there is only one floor. In contrast, the cross-aisle length in warehouse two is much longer than in warehouse one because all aisles are located in succession instead of above each other. Therefore per batch the time to reach the farthest zone from the depot should be walked twice. Consequently the time devoted to picking in warehouse two is defined as follows:

$$\begin{aligned}
 \text{Time devoted to picking}_{\text{warehouse two}} &= \text{Time devoted to picking}_{\text{warehouse 1}} - \text{total number of floor switches} * tf \\
 &+ 2 \cdot \sum_0^s \text{maximum cross aisle time}
 \end{aligned}$$

The comparison of the two warehouses is given in Figure 30. The Figure shows that warehouse one outperforms warehouse two when the floor-switch time is low ($t_v < C$). Meaning that, when the floor-switch time is really low compared to the number of cross aisles, the pick performance is better when the aisles are placed above each other instead of placed in succession. In other words, the extra cross aisle distance required in warehouse two exceeds the extra floor time needed in warehouse one.

On the other hand, when the floor-switch time is high ($t_v > C$) compared to the number of cross aisles, it is better to place the aisles in succession instead of above each other. This can be explained by the fact that the horizontal cross-aisle switch time is less than the time to switch the associated number of floors.

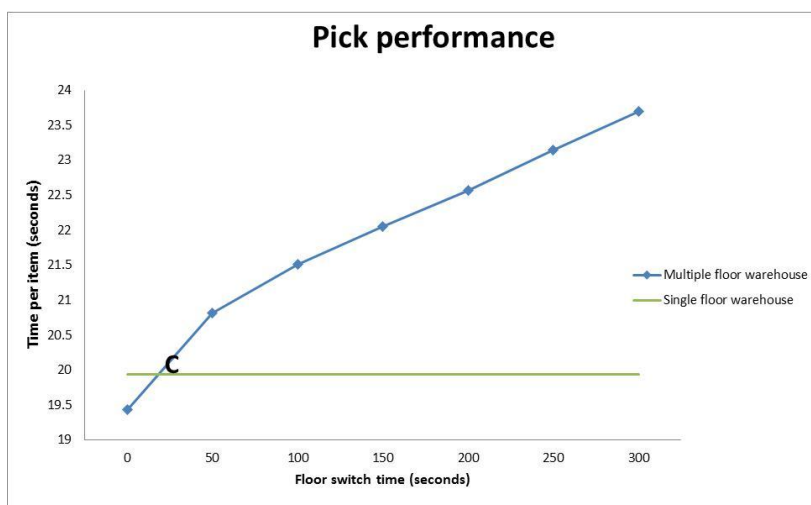


Figure 30 Comparison single floor and multiple floor warehouse

For Docdata, with an associated floor-switch time of 190 seconds it can be concluded, that locating aisles in succession instead of above each other results in a pick performance improvement. Meaning that a floor switch takes more time than the extra cross-aisle time when aisles are located in succession.

10.3 Effect of the S-shape strategy

During this research the S-shape routing strategy is assumed. Hall (1993) already showed that the S-shape strategy is preferred when the numbers of picks per aisle exceed 3.8. For scenario AAB, on average 4.04 picks per aisle are available. This implies that the S-shape assumption performs well on average. However, 72.7 per cent of all aisles have less than four picks. Thus in 72.7 per cent of aisles the S-shape strategy assumption is not preferred. According to Roodbergen and De Koster (2001), the S-shape is most popular in practice. Although the S-shape strategy is not preferred in all cases, it is easy to implement and generates pick sequences which enforce a 'logical' way of working for pickers (Van Nieuwenhuysse and De Koster, 2009). Because the S-shape strategy performs well on average, is easy to implement and logical for pickers this assumption is reasonable.

11. Conclusion and discussion

The aim of this research was to design a method for the pick routing, batch clustering and release policy in a multiple floor warehouse such that the time devoted to picking is minimized. Those three processes have a strong hierarchical dependence, such that a decision made in one phase of the hierarchy strongly influences the decisions that have to be made in the other phases of the hierarchy. Since the decisions made on higher hierarchical levels reduce the design options in the lower levels, the design is made bottom up.

The aim of determining order picking routes consists of finding a sequence in which products have to be retrieved from storage such that travel times are as short as possible. Considering the order picking routes for Docdata, the currently used S-shape routing policy performs well on average. As shown by Hall (1993), this policy outperforms other strategies when the picks per aisle exceed 3.8. The average numbers of picks per aisle in halls 1 and 4 of Docdata are 4.04, for the best performing scenario and thus the S-shape routing policy is recommended according to Hall (1993).

Given that the routing strategy is S-shape, an efficient order batching methodology is a seed algorithm (De Koster et al., 1999). A seed algorithm is composed of two steps. First, the seed (or initial) order has to be selected from those orders not yet added to a route. Second, not yet selected orders have to be added to the seed order until the pick cart is filled to capacity. Next to the current scenario, six scenarios are simulated. Simulation has shown that a seed selection based on the longest processing time (LPT) will outperform the current and the floor longest processing time (FLPT) seed selection rules. In addition, the order addition based on savings or floor savings outperforms the current and LPT order addition rules. The savings order addition adds order j to seed order i that maximize the savings. Savings can be defined as the difference of retrieving the seed order and order j individually compared to retrieving the seed order and order j simultaneously. The floor savings order addition adds the orders to the seed which contain no more floors than the seed order, based on the maximum savings. Which order addition rule is preferred strongly depends upon the time needed for a floor switch. Sensitivity analysis reveals that when the floor switch time is low (< 60 seconds), the savings order addition is preferred. When the floor switch time is medium, the floor savings order addition is preferred. When the floor switch time is high, both order addition rules perform the same, but because of reduced computational complexity, the floor savings order addition rule is recommended. Overall, it can be recommended to Docdata to use the LPT seed selection and the floor savings order addition since the floor switch time is 190 seconds, which can be considered high.

The aim of the release policy at Docdata is twofold: first, all orders should be handled in time and second as efficiently as possible. As shown, a quantity-and-time based releasing policy is recommended to Docdata. A quantity-based releasing policy always outperforms a time-based releasing policy (Chen et al., 2005) and therefore a quantity-based releasing policy is chosen. An increased minimum number of filled pick carts per wave before a wave is released positively influences the pick performance, but negatively influences the due date performance. Considering both pick and due date performance, the cut-off value of the quantity based scheme be at least 0.8 filled pick carts per wave at a wave can be released. The time based frame ensures that all orders are

released timely. After 11 PM no additional orders come in, therefore the time based cut-off value is equal to 11 PM.

More efficient batches can be made when more information is available. A batch can only start when an order picker is available. Because a constant workforce is assumed and the fact that the batch creation time is negligible, it is strongly recommended to Docdata to release batches only when a picker is available.

Overall, this research has shown that a well performing strategy to minimize the time devoted to picking and improve the due date performance for Docdata exists of an S-shape picking strategy, seed algorithm for batching and a quantity-time-based releasing policy. The strategy proposed leads to 17.5 per cent reduction in time compared to the current situation, and has a 100 per cent due date performance.

11.1 Theoretical contributions

Current literature is restricted to batch picking on one floor and vertical movements are disregarded. This research contributes to current literature by designing a methodology to batch over multiple floors and mapping the effect of multiple floors on the pick performance. Sensitivity analysis has shown that an increased number of floors can be beneficial for the pick performance when the floor switch time is low in comparison to the number of cross aisles. On the other hand, locating all pick locations on one floor is more beneficial when the floor switch time is high in comparison to the number of cross aisles.

11.2 Practical contributions

Practical usefulness of the project lies in the possible costs savings and service improvements that can be achieved by changing the releasing and batching strategy. As shown, applying the proposed methodology (scenario AAB) will result in a pick time improvement of 17.5 per cent. In addition, a due date performance of 100 per cent in time batches can be achieved, while the due date performance of the current algorithm is 99.45 per cent. Overall, the proposed methodology improves both the pick and due date performance.

Furthermore, the option to locate all aisles in succession instead of above each other is evaluated. With the current floor switch time, it would have been more efficient to place all aisles in succession instead of above each other. This evaluation can be useful, when the warehouse has to be extended in future.

12.Limitations and recommendations for further research

A number of limitations of this research must be noted. First of all, the whole simulation is executed on a data set of one week. Whether the solution fits the process, depends on the accuracy and representatively of the data. In addition, the scenarios are only executed upon non-peak weeks. For a more complete overview of the performance of the algorithm and in order to take generalizable reliable conclusions, also the peak weeks should be used as input for simulation. Further research should provide more insight into the effect of an increased number of orders on the pick and due date performance.

Second, for modelling purposes, some simplifying assumptions are made. Some assumptions are commonly made in research, and reasonable. For other assumptions relaxation is desirable, and these are mentioned below. First, it is assumed that the routing strategy is S-shape. Whether the S-shape strategy fits the data depends on the number of picks per aisle. Although on the average the S-shape strategy is appropriate, 72.7 per cent of the aisles contain fewer picks than necessary. The savings are computed based on a S-shape routing strategy, and thus also the performance of the batch clustering depends on the routing strategy. Roodbergen and De Koster (2001) show that the combined heuristic is the best heuristic; therefore further research should map the possible improvements of applying a combined heuristic.

The second assumption which limits this research is the assumption that the zone switch time is equal for all types zone switches. However, at Docdata zones can be distinguished which are placed next to each other and above each other. Data analysis has revealed that a horizontal zone switch requires less time than a vertical zone switch. Therefore further research should distinguish between the different types of zone switches, and map the effect on the pick and due date performance. Moreover, the elevator switch time is assumed constant, while it is strongly dependent upon the queue in front of the elevator. When the numbers of orders are increasing, the waiting time in front of the elevator will increase as well. This effect is neglected, and further research should map the effect of an increased waiting time in front of the elevator on the pick and due date performance.

Furthermore, it is assumed that a constant workforce is available during the day, while Docdata can change the workforce during the day. By more flexible assigning the workforce over the day, the proposed solution can be improved.

The batch sizes are currently determined based on an estimation of the pick device capacity. For every wave, the number of items that fit in the pick cart is assumed constant. However, the dimensions of the items are known and not identical. Therefore, it is interesting for further research to indicate the impact of determining the pick cart capacity per wave dynamically.

In this research, the location to pick an order is determined in the order run based on the lowest inventory position. An item can be located at multiple locations and is thus fragmented over the warehouse. Ho and Sarma (2009) define fragmentation as the 'scattering' of identical items throughout the warehouse. By storing identical items at a number of locations, the ordering of pick locations results in different traversal distances and a cost savings opportunity arises. While the

storage allocation can facilitate picking, this saving opportunity is not taken into consideration yet, and this location assignment problem should be solved by further research.

The last limitation of this research is the assumption that the depot is located at corner of the warehouse. While this is the case for Docdata, it reduces the generalizability of the insights obtained. To make the findings more general, this assumption should be relaxed in further research.

Concluding, further research should indicate the effect of the proposed methodology on peak periods. In addition further research could consider relaxing the following assumptions: the S-shape routing strategy, constant floor switch time, constant workforce, constant batch sizes, pick location determination and depot location.

13. Glossary

Automatically article batches:	Batches consisting single orders for a unique article.
Batch:	A set of orders which can be retrieved by a single picking tour (De Koster et al., 2006)
Batching:	(Order) batching is grouping a set of orders into a number of subsets, each of which can then be retrieved by a single picking tour (De Koster et al., 2006).
Batching effect:	Effect which occurs when the numbers of orders are large. Consequently, the waiting time will be small because fewer setups need to be performed. On the other hand, the extraction time and travel time will be large, and will be the dominant aspects of the throughput time (Van Nieuwenhuysse and De Koster, 2009).
Can-go-order:	Orders with a hold date longer than 24 hours.
Catwalk:	A storage area with multiple floors connected by an elevator.
CVRP:	Capacitated vehicle routing problem: the problem of finding a set of minimum cost routes visiting clients with non-negative demands and such that: each route starts and ends at the depot, each client is visited exactly once, and the total load on each route does not exceed the vehicle capacity (Pang and Muyldermans, 2013).
Depot:	The location at which every route starts and ends.
Docdata commerce B.V.:	Division that designs web shops for customers.
Docdata Fulfilment B.V. :	Division that provides the logistic services for the client.
Docdata Payments B.V.:	Division that is responsible for the payments between customer and Docdata NV.
Due date:	Time at which the picking department has to be finished (00.30 AM).
Due date performance:	Percentage of batches that are finished after 00.30 AM.
Equinox line:	This outbound line handles big items which are not 'too big for trolley'.

Floor switch time:	Time that is needed to switch floors including waiting time in front of the elevator. This time is equal for every type of floor switch combination.
Hold date:	The date that the orders has to be handed over to the distributor.
IAI Industrial systems B.V:	A high-tech system builder.
Item:	A product that can be ordered.
Largest gap strategy:	Strategy in which all sub-aisles except the first and last visited are left at the same side as they were entered.
Location type:	Type of storage: catwalk, pallet or release.
Must-go-order:	Orders with a hold date less or equal to today.
NCO orders:	Orders which has to be send retour supplier.
Neopost machine:	This outbound line sorts and packages large items automatically.
Order:	A customer request consisting of one or multiple order lines.
Order addition:	Criteria at which orders are added to the seed order.
Order import:	The process, at which the orders placed at a web shop are communicated to Docdata.
Order line:	An unique product or stock keeping unit (SKU) in a certain quantity.
Order picking:	The process of retrieving products from storage in response to a specific customer request (De Koster et al., 2006).
Order run:	Process that places orders on a pick list.
Outbound line:	The line at which the items are sorted and packed.
Parallel (synchronized) picking:	A number of pickers start on the same order, each order picker is his own zone. The partial orders are merged after picking.
Parts-to-picker systems:	Automated storage and retrieval systems, using mostly aisle-bound cranes to retrieve one or more unit load, and bring them to the pick position.
Picker-to-parts systems:	Systems in which order pickers go through the aisles to retrieve the orders.

Picking:	The process of retrieving products from storage in response to a specific customer request (De Koster et al., 2006).
Picking ratio:	The number of items that can be retrieved per minute.
Pick cart:	The device in which the items are transported during the picking tour.
Pick cart capacity:	The number of items that fit in the pick cart.
Pick device:	Cart with a limited capacity in which the items are picked.
Pick list:	Contains information about one order and the location(s) where to pick this order.
Pick performance:	Time needed per item.
Picking:	The process of retrieving products from storage in response to a specific customer request (De Koster, Le-Duc and Roodbergen, 2007).
POS:	This outbound line sorts single <i>medium sized items and multiple small sized items</i> .
Progressive assembly:	One order picker to start on the order, when he finishes his part, it is handled to the next picker.
Quantity-based releasing:	Releasing methodology, in which a release is dispatched when a certain quantity of outstanding demand is accumulated (Chen et al, 2005).
Saturation effect:	Effect which occurs when the numbers of orders per batch are small. Consequently, the extraction time and travel time are small, but the waiting time will be large. This is due to the fact that, with small values the numbers of orders, setups need to be performed frequently, which causes the utilization of the pickers to increase. Consequently, the batches will need to wait longer in queue before a picker becomes available, which increases the throughput time of the orders (Van Nieuwenhuysse & De Koster, 2009).
Savings algorithm:	Savings algorithms compare the saving in time s_{ij} they can obtain by combining two orders i and j in one route (with pick time t_{ij}) compared to the situation where both orders are collected individual (with order pick time t_i+t_j). Hence, $s_{ij} = t_i + t_j - t_{ij}$.

Set:	A combination of a number of predefined waves.
Seed algorithm:	Algorithm that is composed of two steps. First, the seed (or initial) order has to be selected from those orders not yet added to a route. Second, not yet selected orders have to be added to the seed order until the pick cart capacity is filled.
Seed order:	Initial order of the seed selection heuristic.
S-shape strategy:	The order picker enters every sub-aisle where an item has to be picked and travels the entire aisle.
Start up time:	Time to set up a batch, including receiving empty pick carts and printing batch forms.
Time-based releasing:	Releasing methodology in which a release is dispatched periodically (Chen et al, 2005).
Time-and-Quantity based Releasing:	In this releasing scheme, all orders are hold until one of the following situations occur: a predetermined shipping date or an accumulation of a prefixed quantity (Chen et al, 2005).
URO orders:	Defect orders.
VPM:	Outbound line which sorts and packages <i>single small sized item</i> .
Wave:	Cluster of orders that can be sorted or packed simultaneously.
Wave picking:	The process in which orders for a common destination, in case of Docdata outbound line, are released simultaneously for picking in multiple warehouse areas (De Koster et al., 2006).
24h order:	An order which has to be delivered within 24 hours.

14. Notation

14.1 Variables

- a_f = Number of aisles on floor f
 $a_{i,f}$ = Aisle of the i th pick on floor f
 $a_{js} = \begin{cases} 1, & \text{if and only if order } j \text{ is included in batch } s \\ 0, & \text{Otherwise} \end{cases}$
 c_w = maximum number of orders per batch dependent on the type of wave w
 $da(l)$ = Aisle length of the location type l (Figure 11)
 $dc(l)$ = Cross aisle distance of the location type l (Figure 11)
 d_s = Total horizontal distance to pick all items of the orders in batch s
 d_g = Total horizontal distance to pick all items of order g
 $d_{i,j}$ = Total horizontal distance to pick all items of the orders i and j
 et = Extraction time per item (seconds)
 F = floor number, $f \in F, f = \{0,1,2,3,4\}$
 f_s = Number of floors in batch s
 f_g = Number of floors in order g
 $f_{i,j}$ = Number of floors in orders i and j
 $FPC_{Y,t}$ = The number of filled pick carts per wave y at time t
 i_s = Number of items in batch s
 $n_{s,l,f}$ = Number of visited aisles of batch s on location type l of floor number f
 L = Location type, $l \in L, l =$
 $\{\text{Catwalk hall 10, Catwalk hall 13, Pallet hall 10, Pallet hall 13, Release}\}$
 $RO_{x,t}$ = The run out of outbound line x at time t
 S = the set of all feasible batches, $s \in S$
 st = Startup time of a batch
 s_l = Start aisle of location type l : $s_l =$
 $\{S_{\text{Catwalk hall 10}}, S_{\text{Catwalk hall 13}}, S_{\text{Pallet hall 10}}, S_{\text{Pallet hall 13}}, S_{\text{Release}}\}$
 tf = Floor switch time
 v = Horizontal travel speed
 $x_s = \begin{cases} 1, & \text{if batch } s \text{ is selected} \\ 0, & \text{Otherwise} \end{cases}$

14.2 List of abbreviations

FLPT:	Floor longest processing time
KPI:	Key performance indicator
LPT	Longest processing time
LSP:	Logistic service provider
SKU:	Stock keeping unit
VBA:	Visual Basic for Applications, programming language

15.Literature

- Chen,F.Y., Wang, T. and Xu, T.Z. (2005), Integrated Inventory Replenishment and temporal Shipment Consolidation: A Comparison of Quantity-Based and Time-Based Models. *IIE Transactions* 135;1; p. 197-210.
- De Koster, R., and Van Der Poort, E.(1998), Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions, *IIE Transactions*, 30;5; p. 469-480
- De Koster, M.B.M., Van der Poort, E.S., and Wolters, M. (1999), Efficient orderbatching methods in warehouses, *International Journal of Production Research*, 37;7; p. 1479-1504.
- De Koster, R., Le-Duc, T., and Roodbergen, K.J. (2007), Design and control of warehouse order picking: a literature review. *European Journal of Operational Research* 182;2; p. 481-501.
- Dekker, R., De Koster, M.B.M., Roodbergen, K.J., and Van Kalleveen, H. (2004), Improving order-picking response time at Ankor's warehouse, *Interfaces* 34;4, p. 303-313.
- Dósa, G., (2007). The tight bound of first fit decreasing bin-packing algorithm is $FFD(I) \leq 11/9OPT(I) + 6/9$. In: *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*. LNCS, vol. 4614. Springer, Berlin, pp. 1–11.
- Elsayed, E.A., and Lee, M.-K. (1996), Order processing in automated storage/retrieval systems with due dates. *IIE Transactions*, 28;7; p. 567-577.
- Fleszar, K., Osman, I. H., and Hindi, K. S. (2009), A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 195;3; p. 803-809.
- Fisher, M.L., Jaikumar, R. and Wassenhove, L.N. (1986), A Multiplier Adjustment Method for the Generalized Assignment Problem, *Management Science*, 32;9; p. 1095-1103
- Gademann, N., and Van De Velde, S., (2005). Order batching to minimize total travel time in a parallel-aisle warehouse, *IIE Transactions*, 37;1; p. 63-75.
- Hall, R.W. (1993), Distance approximations for routing manual pickers in a warehouse, *IIE Transactions*, 25;4; p. 76-87.
- Ho, S.S. and Sarma, S. (2009), The Fragmented Warehouse: Location Assignment for Multi-Item Picking, *Logistics and Industrial Informatics*, p. 1-6.
- Jansen, K., and Solis-Oba, R. (2011). A polynomial time $OPT + 1$ algorithm for the cutting stock problem with a constant number of object lengths. *Mathematics of Operations Research*, 36;4; p. 743-753.

Le-Duc, T., & de Koster, M.B.M. (2007). Travel time estimation and order batching in a 2 block warehouse. *European Journal of Operational Research*, 176(1), 374-388. DOI: [10.1016/J.EJOR.2005.03.052](https://doi.org/10.1016/J.EJOR.2005.03.052)

Pang, G., and Muyldermans, L. (2013). Vehicle routing and the value of postponement. *Journal of the Operational Research Society*, 64;9; p. 1429-1440.

Parikh, P. J., and Meller, R. D. (2008). Selecting between batch and zone order picking strategies in a distribution center. *Transportation Research Part E: Logistics and Transportation Review*, 44;5; p. 696-719.

Roodbergen, K.J. (2001). Layout and routing methods for warehouses. Ph.D. thesis, Erasmus University Rotterdam, The Netherlands

Roodbergen, K.J., and De Koster, R. (2001), Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39;9; p. 1865-1883.

Roodbergen, K.J. and Vis, I.F.A. (2006), A model for warehouse layout. *IIE Transactions* 38;10; p. 799-811.

Schneeweiss, C. (2003) Distributed decision making in supply chain management, *European Journal of Operational Research*, 150, Issue 2, 16 October 2003, 237-252.

Tarn, J.M., Razi, M.A., Wen, H.J., and Perez, A.A. (2003), E-fulfillment: the strategy and operational requirements. *Logistics Information Management*, 16;5; p. 350-362

Tompkins, J.A., White, J.A., Bozer, Y.A., Tanchoco, J.M.A. (2003), Facilities Planning, John Wiley and Sons, New York.

van Nieuwenhuysse, I., & de Koster, M.B.M. (2009). Evaluating order throughput time in block warehouses with time window batching. *International Journal of Production Economics*, 121(2), 654-664. DOI: [10.1016/J.IJPE.2009.01.013](https://doi.org/10.1016/J.IJPE.2009.01.013)

Vaughan, T.S., and Petersen, C.G. (1999), The effect of warehouse cross aisles on order picking efficiency, *International Journal of production Research*, 37;4; p. 881-897.

Wolters, M., (1996), Het batchen van orders in een magazijn; een vergelijking van heuristieken. Master's thesis, University of Groningen (in Dutch).

Xu, X., Liu, T., Li, K. and Dong, W. (2014), Evaluating order throughput time with variable time window batching, *International Journal of Production Research*, 52;8; p. 2232-2242