Master Thesis Industrial Engineering and Management

IMPROVING EFFICIENCY IN FULFILMENT CENTRE

UNIVERSITY OF TWENTE.

Author:
P.P.G. Mulder
S2344297

Supervisors
Dr. ir. E.A. Lalla-Ruiz
Dr. B. Alves Beirigo
N. ter Brugge

Improving efficiency in fulfilment centre

Author
P.P.G. Mulder
S2344297

Home institute
University of Twente
Drienerlolaan 5
7577 NB, Enschede

Department
Industrial Engineering and Business Information Systems (IEBIS)

Faculty
Industrial Engineering and Business Information Systems (IEBIS)

Programme
MSc Industrial Engineering and Management (MSc IEM)

Specialisation
Production & Logistics Management

Internal Supervisors
First supervisor: Dr. ir. E.A. Lalla-Ruiz (Eduardo)
Second supervisor: Dr. B. Alves Beirigo (Breno)

External Supervisors
N. ter Brugge (Niels)

Management summary

This research was conducted at Company X, a fulfilment centre located in Enschede. The company is specialised in taking over all of the logistical processes of their clients. The main operations at the warehouse are inbound, storage, outbound and returns. Company X experienced efficiency problems, resulting in high lead times and error margins. These problems mostly occurred in the outbound process, which entails the picking, sorting and packaging of customer orders. The main objective of this research was investigating how technological innovations can improve the efficiency of the order picking operations. The main research question addressed was:

"How can Company X align its order picking processes with industry best practices to improve efficiency?"

To answer this, first a comprehensive analysis of the current warehouse operations was performed. The different processes were investigated and sources of inefficiencies were identified. Furthermore, different KPIs were described and evaluated. In the literature review, different models for solving the joint batching and routing problem were examined, in addition to the impact of innovative warehouse technologies such as barcode scanners.

A two-stage solution approach was designed to solve the Order Picking Problem with Batching and mixed-layout Ultra-Narrow Aisles (TS-OPPB-UNA). For the first stage, a batch formation algorithm was developed to create efficient and dense batches by grouping customer orders. For this stage, it was shown that a Clarke & Wright savings algorithm provided the best batches. In the second stage, a route optimisation algorithm was implemented to find the best route along all pick locations within a batch. An exact Mixed Integer Linear Programming (MILP) model was created to act as a benchmark for the more scalable Simulated Annealing (SA) algorithm. Within the SA framework, a random operator selection strategy yielded the best balance between solution quality and computational time. After extensive tuning using synthetic datasets to ensure robustness, a final algorithm was selected.

By benchmarking the solution approach to the MILP and historical data of Company X, it was found that the SA model provided significant improvements over the current situation. The total number of batches required to fulfil all orders in a given dataset was reduced significantly, which resulted in a reduction of the travel distance of more than 50%. The scenario analysis demonstrated that batch size, technological innovations, and warehouse layout changes can have some significant impact on the total travel distance and picking time.

To ensure practical impact for Company X, the thesis emphasises that the implementation of the solution approach should be combined with digital picking technologies to guide order pickers through the optimised routes, ensuring that they follow the routes precisely. Additional recommendations include the recording of product volumes and cart capacities, so that batches can be formed more accurately.

Preface

Dear reader,

You are about to read my MSc thesis named 'Improving efficiency in fulfilment centre'. This research was carried out at Company X in Enschede as the final requirement for the completion of the Master's program in Industrial Engineering and Management at the University of Twente, with a specialisation in Production and Logistics Management.

During my Master's program, I have gained extensive knowledge on various topics, which I could apply in this research to address a complex, real-world problem. While working on this project, I deepened my understanding of warehouse operations, which was very interesting. Therefore, I would like to thank all the colleagues of Company X for their interest and for creating a welcoming and enjoyable work environment. In special, I would like to thank Niels ter Brugge for his supervision, his practical insights and the company specific knowledge I received throughout the project.

Furthermore, I would like to express additional gratitude to my UT supervisor, Eduardo Lalla, for his guidance, insightful feedback and continuous support throughout this project. I would also like to express my appreciation of the involvement of Breno Alves Beirigo, who acted as my second supervisor.

Finally, I would like to thank my family and friend for their support and encouragement, both throughout this final phase and my entire study period. I have enjoyed the years of studying at the University of Twente.

I hope that the results presented in this thesis will be valuable not only to Company X, but also to others seeking to improve warehouse efficiency.

Pim Mulder
Enschede
August, 2025

Table of contents

1. Introduction

This chapter covers the introduction to this research. In Section 1.1, the company 'Company X' is introduced. Section 1.2 provides a description of the identified problem. In Section 1.3 the problem goal is outlined. In Section 1.4, the research questions and sub-questions are formulated. Finally, in Section 1.5, an overview of the research design is given.

1.1 Company introduction

Company X is a fulfilment centre located in Enschede. Operating from a warehouse with a size of approximately 3500m^2, they help more than 30 clients with the processing of their orders. The clients of Company X are both other businesses (Business to Business, B2B) and customers (Business to Clients, B2C). Since 2023, Company X is part of the Group Y, a large logistical service provider in Twente, specialised in all sorts of logistical processes; from (inter)national road transport and sea freight to air freight and warehousing.

Company X specialises in taking over all of the logistical processes of their clients. Upon arrival of products at the warehouse, the products go through the inbound process, which includes unloading the trucks and the receiving and processing of the goods to ensure accurate inventory management. Once the incoming goods are processed, they are stored in designated locations in the warehouse, until customer orders are placed. When an order is received, the outbound process starts. First, the required products are picked and sorted, after which the order is packaged and prepared for shipment. The orders are then distributed to customers by various carriers. When customers are unsatisfied with their purchase, a structured return process is in place to facilitate the return of goods to the warehouse. Once there, the goods are subjected to a quality inspection. If they meet the required standards, they are returned to their storage location in the warehouse. If not, they are written off. In Figure 1.1, an overview of the logistical processes is shown.

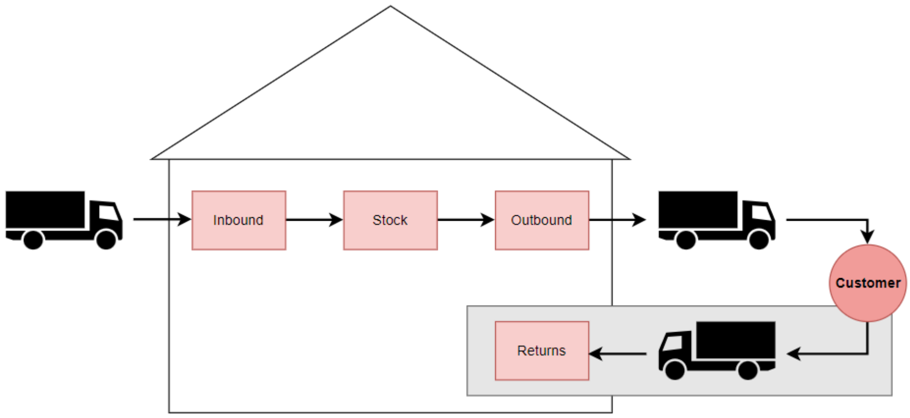


Figure 1.1 Overview of logistical processes at Company X

1.2 Problem description

To identify the core problem of Company X, the Managerial Problem-Solving Method (MPSM) of Heerkens and Van Winden (2017) has been used. This process started with making an inventory of existing problems, after which the causes and effects are visualised in a structured problem cluster.

In the current situation, Company X experiences different inefficiencies in the warehouse, leading to 'high error margins' and 'high lead times'. Since there is a discrepancy between the norm and reality (both the error margin and lead times should be low but currently are not), these problems are considered action problems. The action problems result in ‘low customer satisfaction’ and ‘high operational costs’, making these the consequences of the action problems.

The high lead times are largely attributed to two internal issues: a lengthy order picking process and delays in inbound processes. The latter is driven by factors outside of Company X’s control, since some suppliers are providing incorrect products or documents, resulting in additional actions when these events occur. The high error margin is caused by a high chance of human error. This leaves two actionable internal challenges, namely a picking process that takes too much time and a high chance of human errors.

At the root of these challenges lies the problem of a suboptimal picking route, which is identified as the core problem of this research. Inefficient routes directly contribute to lengthy picking times and increases the risk of human error.

While other factors, such as storage assignment, warehouse layout, and a general lack of insight into technological innovations also contribute to inefficiencies, they are not the main focus of this study. These issues are broad and strategic in nature, making them less suitable for the main focus of the research. However, concentrating on picker routing allows for the development of a concrete and operational solution, directly impact performance. By investigating how specific technological innovations can be used to support and enhance the routing process, more insight is provided about potential innovations that can be implemented at Company X.

Instead, by focusing on optimising the picker routes, a more tangible problem was found that can have practical impact directly. Improvements made on a picker route can be evaluated immediately through key performance indicators (KPIs) such as travel time and error rates. Importantly, it should be noted that this narrowed focus does not exclude technological innovation. It rather allows for a targeted investigation into how technology can be integrated with or replace the current order picking process.

In Figure 1.2, an overview of the different causes and effects mentioned above are shown in a problem cluster.

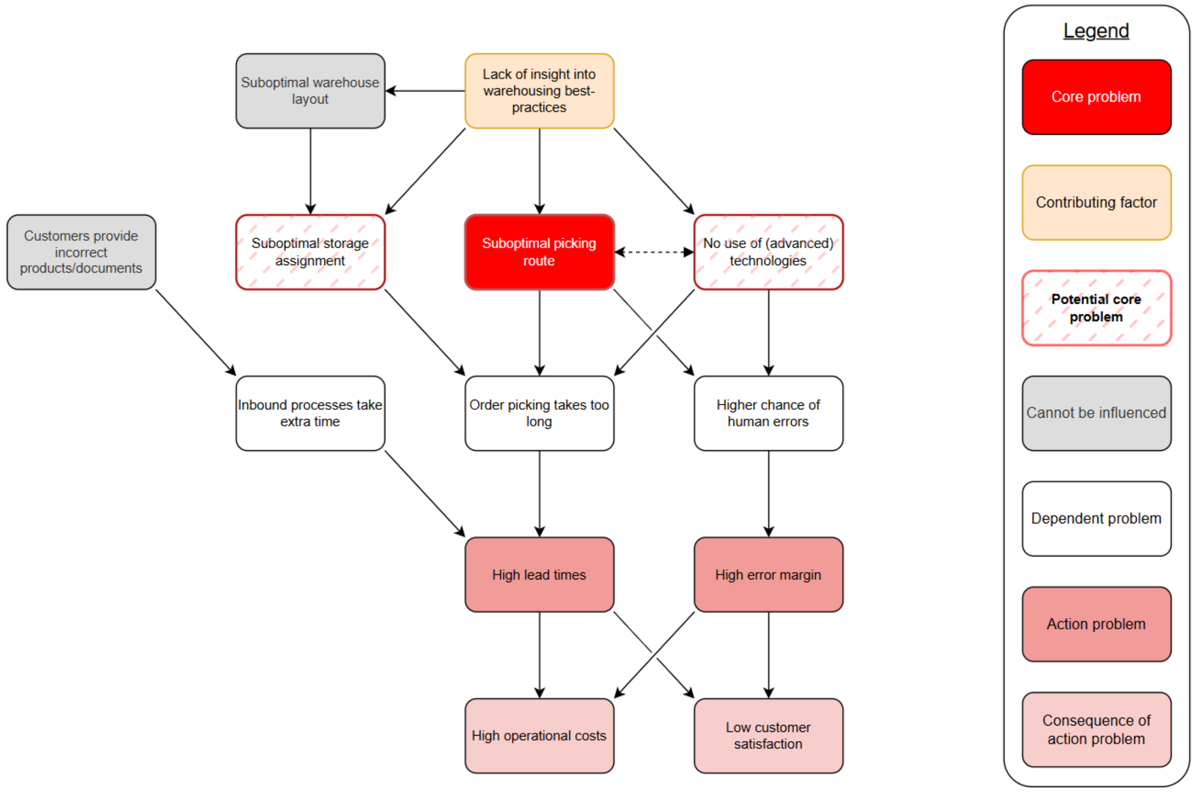


Figure 1.2 Problem cluster

1.3 Research goal

The goal of this research is to improve the operational efficiency of Company X by addressing optimising the current order picking process, particularly focusing on the picking routes within the warehouse. By reducing the travel time and minimising the chance of errors, this study seeks to make improvements in lead time and accuracy.

To achieve this, the research involves the development of a mathematical algorithm capable of determining the most efficient picking routes. The algorithm is tailored to Company X's warehouse layout and order patterns, aiming to minimise travel time and distance during the picking process. With the optimised picking routes, it is expected that the lead times are reduced and error margins decreased.

In addition to route optimisation, this research will also investigate how technological innovations could enhance or support the picking process. Instead of proposing the replacement of the current picking process, the aim is to find innovations that may be suitable for integration and under what conditions they provide additional value. This keeps the primary focus of the research on the operational optimisation, while it also identifies potential directions for future innovation.

Research scope

This research focuses on the optimisation of the order picking process within Company X's warehouse. It specifically addresses the routing of order pickers to minimise travel times and reduce the chance of errors. The scope includes:

- Analysis of current warehouse operations
- Development of an optimised routing approach, designed for Company X's warehouse
- Evaluation of the solution using operational KPIs

- Exploratory assessment of technological innovations for the picking process

To further narrow down this research, it was decided that the technological innovations are assessed only in relation to the picking process. This means that innovations for other warehouse operations like receiving or packaging are not considered. While the research highlights potential technological innovations, the implementation or development of hardware is outside the scope of this study.

Furthermore, factors that cannot be influenced by Company X are out of scope of this research. These include, amongst others, the inaccurate documentation provided by suppliers and the layout structure of the warehouse, since these cannot be changed without incurring extreme costs or operational disruptions.

Research limitations

This research has several limitations. First, the findings are specific to and based on the operational environment of Company X, potentially limiting generalisability to other warehouses. Factors influencing this operational environment include the static warehouse layout, the number of employees and the product assortment. Second, the optimisation model is based on a simplified representation of the warehouse and substantiated assumptions, due to data and time constraints. Third, it is assumed that the storage assignment is static, meaning that it does not change over time. While this is generally true, there are occasional reassignments of products, although this does not happen too frequently (typically about once a month for only a few products). Finally, the available data may not capture all relevant aspects of the order picking process, like picker preferences and habits.

1.4 Research questions

The aim of this research is to answer the main research question, which is formulated as follows:

"How can Company X align its order picking processes with industry best practices to improve efficiency?"

To give answer to this main research question, additional research questions have been defined, such that the main problem can be solved in smaller, systematic steps. By answering these research questions, the core problem is solved, which in turn results in the solution for the action problems. In total, five research questions have been defined, each corresponding to a specific section of the research. These sections and research questions are explained below.

Context Analysis

The first research question aims to analyse and understand the current situation. It involves collecting information on how the warehouse operates, the key factors influencing the current routing policy such as the storage assignment and batching strategy, and the technological infrastructure used in the warehouse. Additionally, relevant KPIs, constraints and requirements are examined. The first research question and its sub-questions are formulated as follows:

1. How does Company X currently operate?
    a) What are the different processes at Company X?
    b) What factors influence the current routing policy?
    c) What are technologies currently supporting the warehouse?
    d) What are the KPIs, constraints and requirements of the picking process?

Literature review

To develop an optimised order picker routing policy, first information has to be gathered on existing strategies, models and technologies that have been proposed in literature. This is done in the second part of this thesis, where relevant literature on routing policies, algorithmic approaches, and supporting technologies are examined. This provides answers to the second research question and sub-questions, which are formulated as follows:

2. Based on the literature, what approaches and innovations can be used to optimise or transform order picker routing processes?
    a) What strategies and approaches are described in the literature for optimising routing policies?
    b) What computational methods are available in the literature to solve order picker routing problems in warehouses?
    c) What technological innovations are available to support or transform warehouse order picker processes?

Design of solution approach

The next step in the research of improving order picker routing is to design an optimised solution approach. In this third section, the routing problem is formulated based on warehouse specific characteristics, an appropriate computational method is selected, and the applicability of emerging technologies is considered. The following research question and sub-questions guide the design process:

3. How can a solution approach be designed to improve the efficiency of Company X's order picking process?
    a) How can the picker routing problem at Company X be formulated considering warehouse specific characteristics?
    b) How can a solution approach be designed to address this problem?
    c) How can relevant technological innovations be incorporated within the solution approach to support long-term improvement?

Experimentation and evaluation

Once the solution approach has been designed and developed, the performance of the model must be evaluated to determine its effectiveness. Various experimental scenarios have to be tested and evaluated to assess how the solution improves warehouse efficiency. The research question below structures the content in this part of the thesis:

4. How does the developed solution improve warehouse efficiency compared to the current situation?
    a) What experimental scenarios should be considered?
    b) How does the proposed solution perform compared to the current situation in terms of costs, travel distance and lead times?
    c) How does the proposed solution perform under different workload scenarios

Conclusion

The final section summarises the key findings of the research and recommendations for Company X are presented. Additionally, limitations of the research are discussed. The following research question and sub-questions guide this section:

5. What conclusions can be drawn and what recommendations can be made based on the research?

a) What are the key findings and outcomes of the research?
b) What recommendations can be made for Company X based on the findings?
c) What are the limitations of this research?

1.5 Research design

The research is structured in several phases, each contributing to solving the research problem. Each phase aims to give answer to one of the research questions stated in the previous section, so that eventually, the main research question can be answered. In Figure 1.3, a visual overview is provided of how each research question is addressed through the different phases of the research, highlighting the required inputs, expected outputs, and their corresponding chapters.

In the first phase, the problem context is identified, where company data and employee experiences are used to understand the current situation at Company X. The literature review, which is the second phase, explores existing routing strategies, optimisation methods and technologies used to support the order picking process. In the next phase, a solution approach is developed by formulating a structured optimisation model. The fourth phase involves testing and evaluating the developed solution under various scenarios. Finally, the conclusions and recommendations of the research are provided.

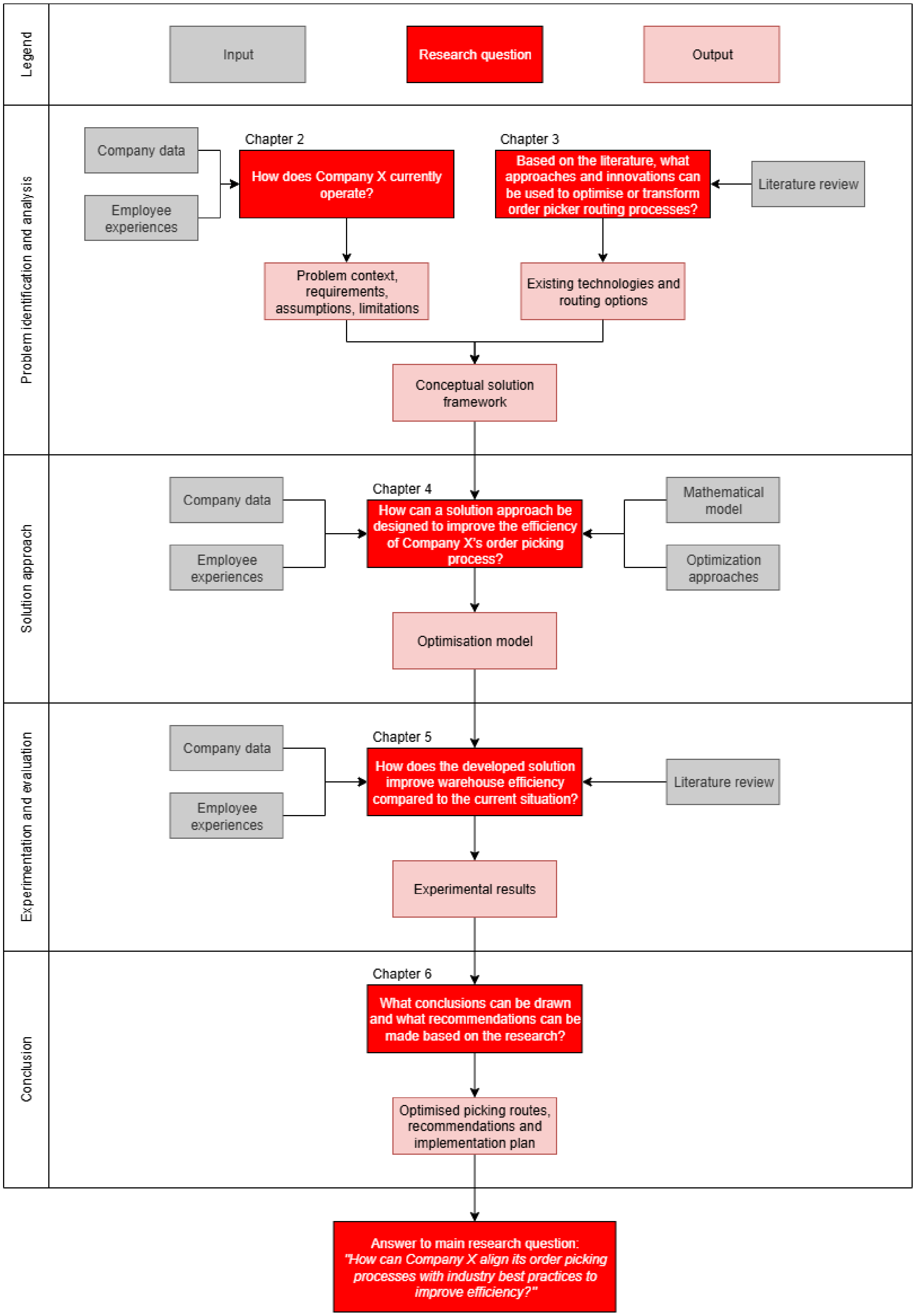


Figure 1.3 Graphical representation of research design

2. Context analysis

In this chapter, the context in which this research takes place is analysed. This gives answer to research question 1, which states: *"How does Company X currently operate?"*. For this, an overview of Company X is given in Section 2.1. In Section 2.2, the order picking process is discussed in more detail. In Section 2.3, the KPIs, constraints, and requirements of the warehouse processes are outlined.

2.1 Overview of Company X

This section provides an overview of the warehouse operations of Company X, outlining how their warehouse is structured, how goods are handled and what technologies are used during the different processes. Additionally, the characteristics of customer orders and demand patterns are explored, offering insight into the volume, size and timing of orders.

2.1.1 Warehouse layout

The layout of Company X's warehouse is organised into different zones that each has their own function. Figure 2.1 shows the layout overview of the warehouse.

The inbound area (purple) contains receiving docks, where the arrival of goods is handled. The majority of the warehouse space is used for storage, where different zones are used:

- Pallet racks (green): These racks accommodate picking and bulk storage.
- Shelves (orange): Smaller items are stored in open shelves for easy access.
- Coarse goods area (yellow): Large or irregular items are stored in open floor spaces distributed across the warehouse.
- Office locations (blue & red): Locations near the office are used for slow-moving products.

At the depot (grey), multiple package tables are placed where order pickers start and finish their picking tour. Here, the orders are packaged and made ready for delivery. Once ready, they are temporarily stored at the delivery zone (dashed).

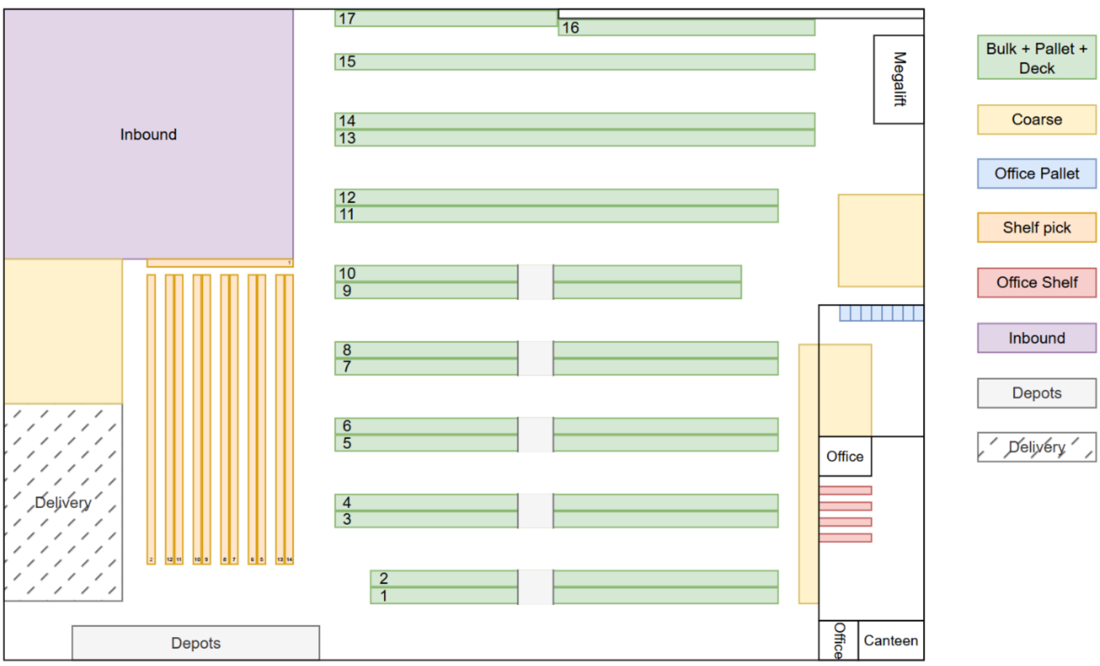


Figure 2.1 Layout overview of warehouse

2.1.2 Warehouse processes

In this section, all the different processes at Company X are explained. To support the explanations, a flowchart is added for each process. Here, a dark grey rectangle represents an automated action or process carried out by the Warehouse Management System (WMS). A light grey rectangle indicates a manual or actionable process performed by warehouse staff. Lastly, a red diamond denotes decision points, where a choice needs to be made before proceeding to the next step.

2.1.2.1 Inbound

The first process at Company X is the inbound process. Figure 2.2 shows a flowchart of this process, which begins when incoming goods are registered by a customer in advance of delivery. The main inbound shipments are supplier deliveries and replenishment stock, typically delivered on pallets.

Upon arrival, the goods are unloaded from the supplying trucks at the designated receiving dock, where they are checked for quantity and condition. The goods are scanned and compared with the expected delivery data. If the goods are accepted, they are assigned storage location and registered in the WMS. When new products arrive, the barcode is stored for tracking purposes. If goods are not accepted, the customer and suppliers are contacted. Based on the agreements they make, the products are processed further.

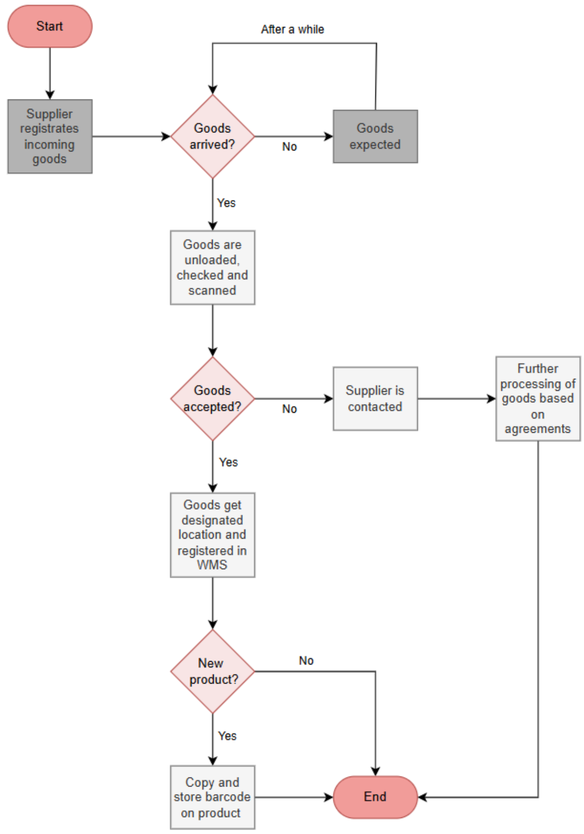


Figure 2.2 Flowchart of inbound process

2.1.2.2 Storage

Once the products are registered and accepted at the inbound, they can be stored away, which is done in the storage process. A flowchart of the storage process is shown in Figure 2.3. The incoming goods are placed into appropriate storage locations to optimise space utilisation and allow for efficient retrieval of goods. Three main types of storage areas are used, namely pallet racks, shelves and coarse goods areas.

- Pallet racks are the primary storage system, consisting of multi-tiered racks. At the ground level, products are accessible, allowing for them to be picked. The higher tiers are used for bulk storage, from which pick locations are replenished.
- Shelves are used for the storage of smaller items. The shelves are located close to the depot to ensure shorter travel times during picking.
- Coarse goods are stored at designated open locations throughout the warehouse, since they are too big for standard racks or shelves.

Some incoming goods are first stored at the bulk locations on the higher tiers of the storage racks. This mostly applies to the goods stored on pallets, although some smaller products, normally stored in the shelves, arrive in larger quantities and thus are also stored at the bulk locations. Once the inventory levels at the picking locations reach a critical level, these goods are retrieved from the bulk locations to replenish the pick locations.

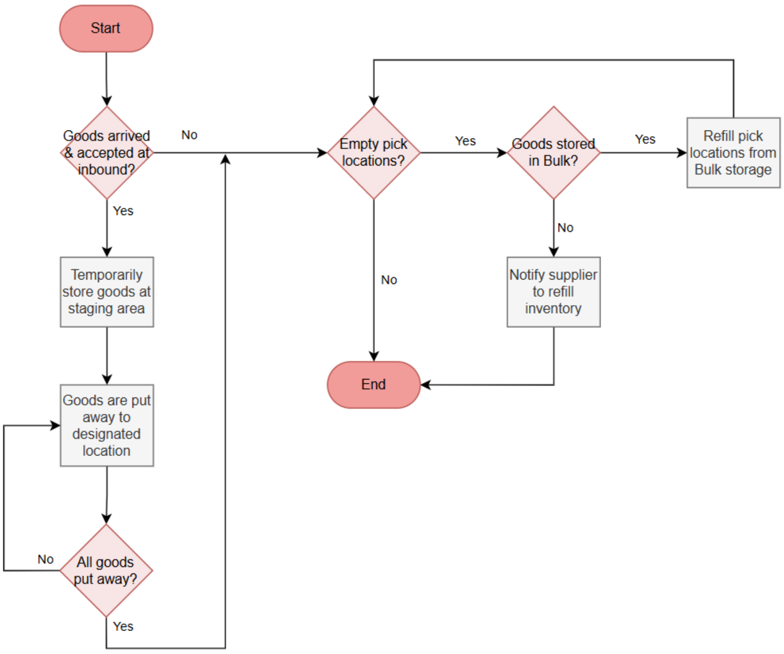


Figure 2.3 Flowchart of storage process

2.1.2.3 Outbound

The outbound process at Company X is related to the collection and packaging of products, after which they are ready to be shipped. In Figure 2.4, the flowchart of the outbound process is shown, which begins when a customer order is received. An automatic check is done to confirm that enough products are in stock, after which a pick list is created for the incoming order. When enough pick lists are available, an employee can let the WMS make a batch, which then is ready to be picked. The whole picking process, including the creation of batches, is discussed in more detail in Section 2.2, which includes the planning process of the order picking.

After all items have been collected, the order picker returns to the depot where the goods are packaged and made ready for shipment. Each order is processed individually to sure that the correct items are packed in appropriately sized boxes. Fragile goods are protected using wrapping and filling materials.

After the products are packed, a label with tracking barcodes and shipping details is generated by the WMS for each package. After that, the packaged orders are stacked on storage carts, where they are temporarily stored until the scheduled delivery. The final step in the outbound process involves helping with the loading of trucks, ensuring that all orders are dispatched on time.

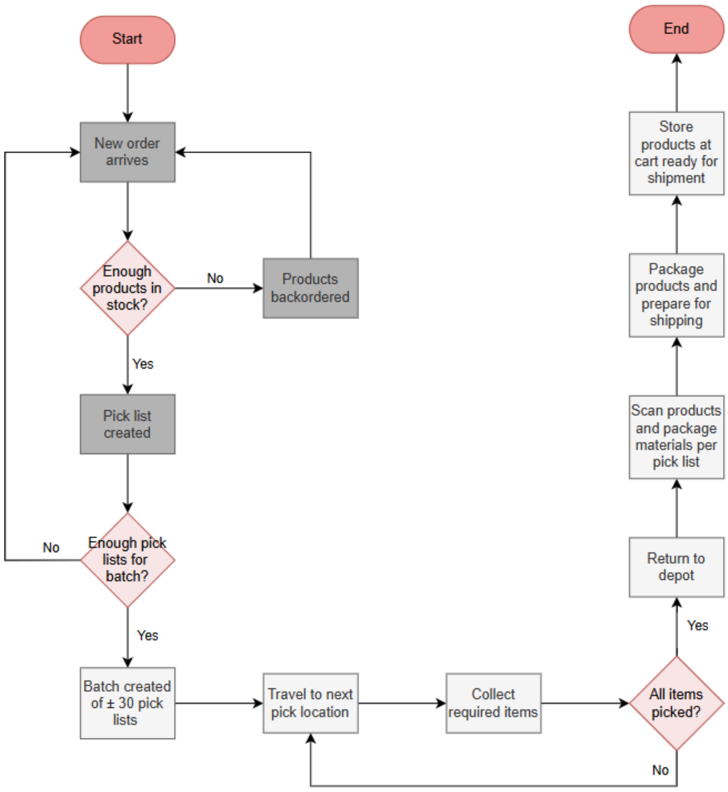


Figure 2.4 Flowchart of outbound process

2.1.2.4 Returns

When a customer is not satisfied with their order, they can return the products. Figure 2.5 shows the return process graphically. When a returned product arrives at the warehouse, it is first inspected for quality and condition.

If a product is in an acceptable condition, it is re-added to the digital inventory through the WMS before it is brought back to its designated picking location. Here, it is stored again, making it available for future order fulfilment. If a product is not in an acceptable condition, it is thrown away.

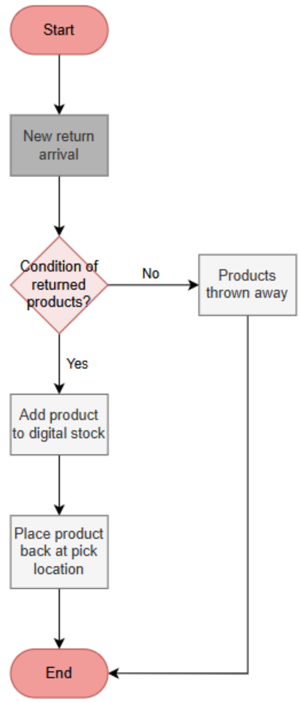


Figure 2.5 Flowchart of return process

2.1.3 Supporting technologies

To support the operational processes described above, various technologies are currently used. Some of these technologies play a crucial role in keeping the warehouse operating, like the warehouse management system and material handling equipment.

2.1.3.1 Warehouse management system

Company X uses a Warehouse Management System (WMS) as their software system to coordinate and oversee warehouse operations. The system is designed to support warehouses and fulfilment centres with, amongst others, inventory tracking, order processing and task prioritisation. By integrating real-time data updates with digital inventory management, warehouse processes are streamlined, improving overall efficiency.

The real-time inventory tracking enables continuous monitoring of stock levels across all storage locations. When goods are received, moved, picked or shipped, an update of the inventory is made, allowing for accurate stock levels and fast replenishments. This reduces the risk of stockouts, while also optimising the use of storage space.

In addition to inventory tracking, the WMS also facilitates order management by generating picklists and combining them into batches. The system allows order pickers to change the priority of tasks based on urgency of orders or availability of goods. This enables dynamic workflow adjustments to meet changing demands.

By providing detailed information about incoming shipments, the system also supports inbound logistics. This enables the inbound department to plan storage allocations in advance, minimising congestion at the receiving docks and reducing processing times.

Returns are also handled by the WMS, allowing for the tracking, assessment and reintegration into inventory of returned goods. Additional processing of returned goods is also supported by the software.

2.1.3.2 Material handling equipment

To facilitate the transportation and handling of goods within the warehouse, Company X uses specialised material handling equipment. The two main types of electric vehicles use are a standing electric pallet truck and an electric reach truck.

The standing electric pallet truck is used for low-level transportation of pallets. The truck can pick up standardised pallets from the floor and transport them to other locations in the warehouse. It is used for loading and unloading of shipments, transferring stock between storage locations, and replenishment of pick locations from bulk area. Since operates are standing on the truck, they can easily move on and of the vehicle.

An electric reach truck is used for the vertical storage and retrieval operations in the pallet rack area. The truck is equipped with an extendable lifting mast which allow operators to access pallets that are not stored at ground level. At Company X, this truck is used to access the bulk storage located on the second and third tier of the pallet racks. This equipment is essential for the warehouse operations, since without it, the higher tiers of the racks were not available, thus reducing space utilisation significantly.

2.1.3.3 Vertical carousel

The warehouse of Company X is equipped with a vertical carousel system, although it is not currently used. A vertical carousel is an automated storage and retrieval (AS/R) system that can rotate vertically to bring stored items directly to a point where they can be accessed. The system is well-suited for storing and retrieving small to medium-sized items that are frequently ordered. Vertical carousels can improve space utilisation and reduced picking time when correctly used. However, due to the current location of the system and the lack of appropriate goods, the vertical carousel is currently not being used.

2.1.4 Order characteristics and demand patterns

As a fulfilment centre, a lot of different customer orders come in at Company X, varying in order volume, frequency, and composition. Understanding the order characteristics and demand patterns is essential for the evaluation of the warehouse performance.

2.1.4.1 Order Volume & Frequency

At Company X, a high volume of orders is processed across a diverse customer base of more than 30 customers. On average, about 225 orders arrive per day, with peaks in the middle of the week. Certain clients of Company X generate substantial number of orders, with two clients generating over 150 orders per day combined. In Figure 2.6, the total number of orders per day are shown, in addition to the shipped orders per day. As can be seen, there are different spikes in the number of orders that are shipped. This is because Company X does not operate in the weekends, so the orders that have accumulated over the weekend need to be processes and shipped at the start of the workweek. Where the shipped orders have a weekly cycle, the incoming orders follow a more evenly distribution. However, there are still some fluctuations, mostly between the weekdays and weekends.

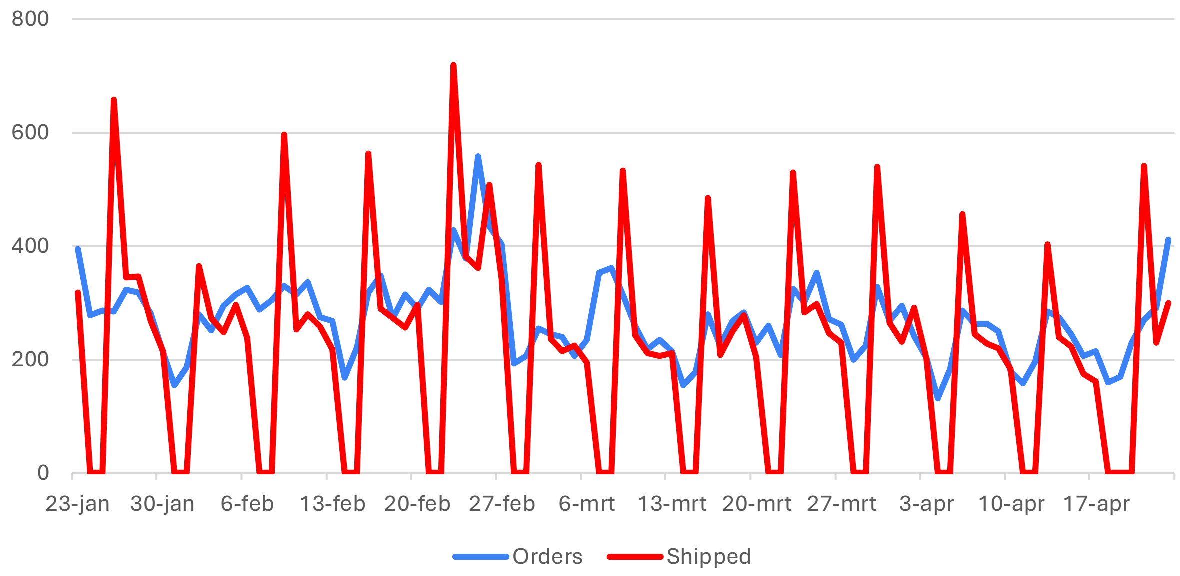


Figure 2.6 Total orders and shipments per day

In Figure 2.7, the average number of orders and shipments per day of the week are shown. Here, it can also be seen that, on average, there are more incoming orders during the weekdays than in the weekend. Additionally, it can be seen that the average shipped orders are relatively constant throughout the workweek, with the exception of the Monday, on which the orders that came in in the weekend have to be shipped.

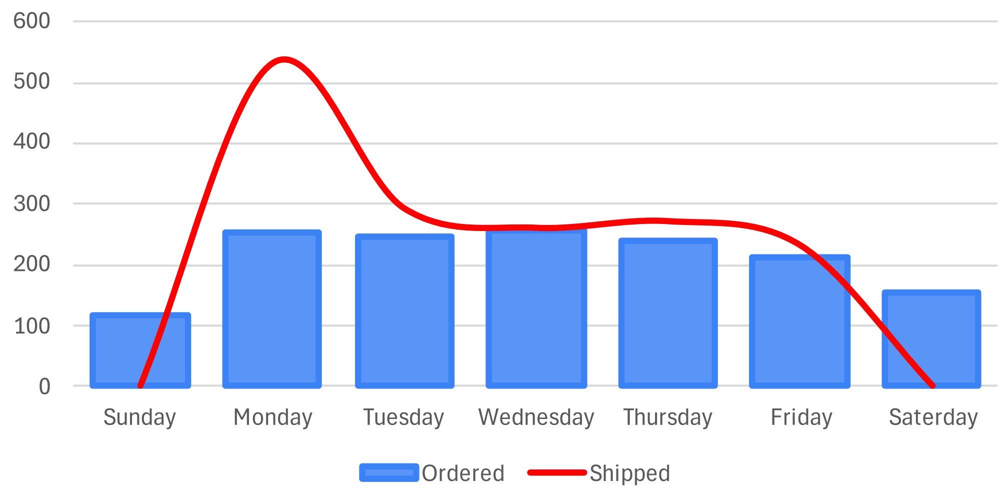


Figure 2.7 Average orders per day of the week

Average hourly data in Figure 2.8 shows that most orders come in during the daytime at a relative steady rate of about 15 incoming orders on average per hour. Over the whole day, the average is about 10 incoming orders per hour. The shipped orders have quite some fluctuations, which can be explained by employee breaks (10th hour, 13th hour and 15th hour) and the fixed time windows of the carrier companies.

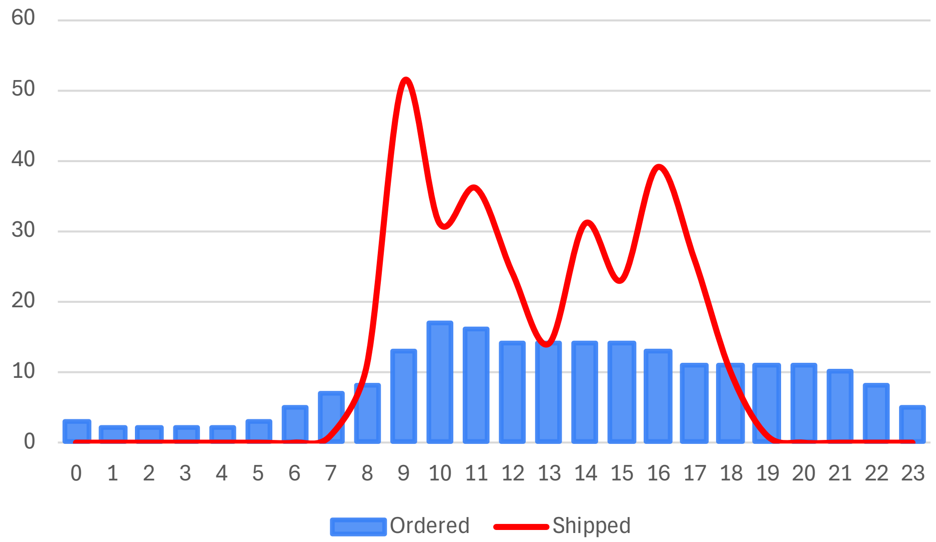


Figure 2.8 Average orders per hour

2.1.4.2 Order Size & Composition

At Company X, the order sizes vary substantially. Some orders only require one item to be picked, whereas others contain hundreds of items resulting in bulk shipments.

The biggest fulfilment client receives about 100 orders per day on average, with each order having an average order size of 9 items. This means that on average, about 900 items have to be picked for this client, which equals about 32% of all items picked per day. In practice, this total lies a bit lower, due to occasional large orders that get picked in bulk.

Another client receives orders with an average of about 23 items per order, while receiving an average amount of 50 orders per day. So on average, about 1150 items have to be picked per day to fulfil orders for this client. This equals about 40% of all items picked per day, while the orders contribute around 20% of the daily orders.

One fulfilment client receives orders that on average exceeds 900 items per order. Such an order comes on average once every three days, so even though they represent only a small share of the total incoming orders (just 0.1%), this client still contributes significantly to the total number of items picked (11.3%).

Conversely, there are also many clients that only receive small orders, with an average of 1 to 5 items per order. Some of these clients receive relatively many orders (more than 15 orders per day on average), while others only receive an average of one order per month.

2.1.4.3 Demand Fluctuations

The demand of Company X shows some fluctuations between the weekend and weekdays. During the weekend, the number of orders is lower than during the week. The main reason for this is that during weekdays, many clients offer a ‘Order today, receive tomorrow’ policy, while this policy is not offered during weekends. Furthermore, some clients experience seasonal demand fluctuations, but these are highly dependent on the clients themselves. For instance, one client that sells carnival clothing has significantly higher demand during carnival or Halloween, whereas the demand decreases right after these festivities.

During the weekdays, the demand is relatively consistent, which enables some predictability for staffing. However, flexibility remains necessary for the occasional arrival of large orders.

2.1.4.4 Service Level Expectations

To ensure the timely delivery of products to customers, Company X works with various carriers. Each of these carriers has a different pickup time, based on the type of delivery service they provide. These different pickup times influence the daily workflow and prioritisation of customer orders.

In Table 2.1, the pickup times of the different carriers are shown. Two carriers provide same day delivery, which requires them to pick up the products early in the day. This means that these customer orders need to be processed the earliest, ensuring that they are ready to be shipped when the carriers arrive. Once the carriers have picked up all required packages, the priority of Company X shifts to customer orders that are picked up later in the day by the other carriers. The carriers are obliged to take all packages that are ready for pickup, so the capacity of the carrier trucks is not a constraint for Company X, merely for the carriers themselves.

Table 2.1 Carrier pickup times

Carrier	*Approximate Pickup Time*	*Additional Info*
VVB (BOL SSD/DHL)	10:00 - 12:00	Packed before 10:00, delivered same day
Trunkrs	10:00	Same day delivery
Cycloon	16:00	—
VVB (BOL) POSTNL	18:00 - 20:00	Packed before 18:00, delivered next day
GLS	17:30 - 18:00	Packed before 17:00
POSTNL (Innosend)	18:00	—

The varying time windows of the different carriers demand tight coordination between the different outbound processes. Especially to facilitate the same-day deliveries, the warehouse staff must prioritise order batches accordingly throughout the day, ensuring that the carriers' service level requirements are met.

2.2 Order picking process

The order picking process typically accounts for about 55% of the total warehouse operating costs (Bartholdi & Hackman, 2019), making it the most critical part of the outbound operations. The picking process involves manually retrieving items from various storage locations throughout the warehouse, playing a key role in determining overall warehouse efficiency and accuracy. In this section, the current picking procedure, routing logic, storage assignment, and picking challenges at Company X are discussed.

Batching

When an order arrives in the WMS, a picklist is created, given that enough products are present. If not, the items are backordered, meaning that the supplier needs to deliver more products before the customer order can be fulfilled. When enough pick lists are in the system, an order picker can create a batch, typically consisting of about 30 pick lists. Currently, the batching logic is based on the courier responsible for pickup. This ensures that all orders in a batch fall within the same pickup time window. Once a batch is created, a list with all required products is printed. No digital devices are used throughout the picking process, meaning that order pickers are entirely reliant on paper lists.

Picker routing

Once an order picker has printed a batch list, they can begin their picking route by taking a manual picking cart. While the route is generally sequential, there is no formal policy or method used for the creation of the routes. Consequently, pickers determine their own routes, often based on convenience or habit, leading to variability and inconsistencies in the picking routes.

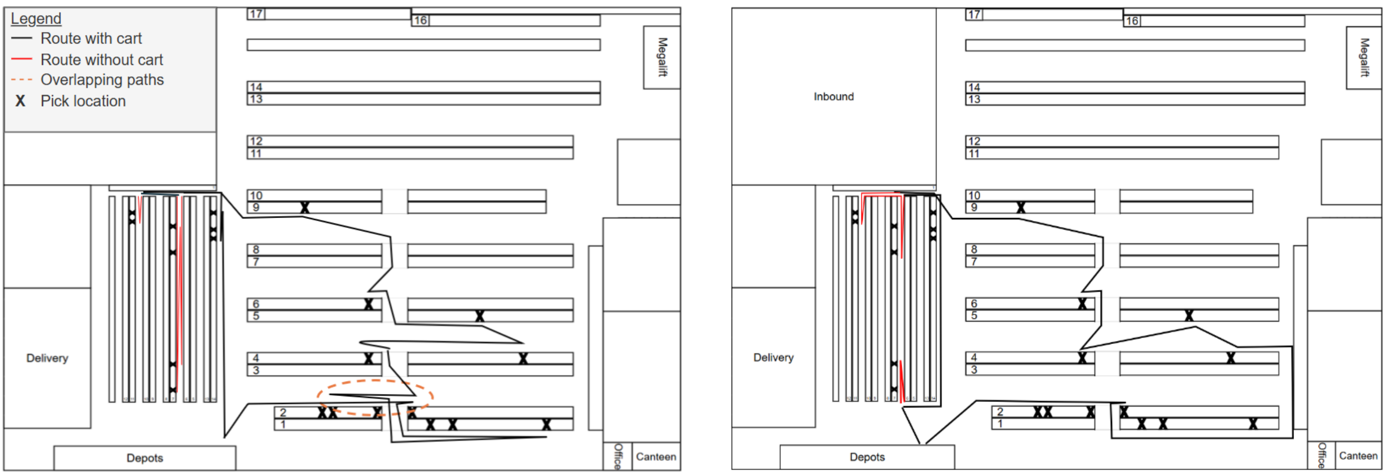
Currently, the products are sequenced based on the (Dutch) alphabetical order of the location names, resulting in the following structure:

1. Deck pick: In aisles 2 and 3, an additional tier is created, known as the *deck*, and it is used for small products that are too small for full pallets but too big for the shelves.
2. Coarse pick: The coarse goods are stored throughout the warehouse, but are not required to be picked in order, due to the nature of the goods. Since these products are often too big for a pallet, they do not fit the picking cart and thus are picked separately.
3. Office pick: Near the office, there are some additional pick locations. Different products are stored here in three location types: Office coarse, Office pallet, and Office shelf. In general, these products are slow-moving, thus not required too often.
4. Pallet pick: The main picking area of the warehouse, consisting of 17 rows of racks with pick locations at the ground level. Between the first six aisles, there is a cross-aisle in the middle where order pickers can travel between aisles. All aisles are accessible from the front and the back cross-aisle
5. Shelf pick: The last picking area, closest to the depot, where small items are stored. Here, twelve rows of shelves are arranged closely together. This requires the picker to leave the picking cart near the entrance of each aisle before entering, retrieving the items manually before returning to the cart.

At each stop, the picker checks the printed batch list, collects the required number of items and places them on the picking cart. After confirming the picked items, the picker moves to the next location. Once all goods are picked, the picker moves back to the depot.

Route visualisation

To show that the current picking approach is inconsistent and can result in high variability of travel distances, two different picking routes are shown for the same pick lists. One follows the default order provided on the printed batch list, while the other follows a route that is manually improved to reduce the total distance travelled. The routes are shown in Figure 2.9, where a visited pick location is represented by a black cross. A black line indicates a route where a picker moves together with their cart, whereas a red line represents a picker movement without the cart. The pick list that was used to create this route is shown in Appendix A.



(a) (b)

Figure 2.9 In (a) the default order route (210 meters) follows the default picklist order, resulting in many overlapping paths and a full pass through one shelf aisle. In (b), the improved order route (150 meters) follows an optimised sequence, reducing the backtracking and the full traversal of the shelf aisle.

Route 1 – Default order

The default route follows the sequence of pick locations as printed on the batch list. While easy to follow, several suboptimal aspects result in an inefficient route:

- In some cases, the order picker passes storage locations without collecting the items stored there, as these locations are sequenced later in the alphabetical ordered pick list. An example can be seen in the red dotted oval in Figure 2.9(a), where all Deck locations in Aisle 2 are picked first. Then, the picker moves to Aisle 1 to retrieve items from the Pallet locations, only to return to Aisle 2 afterward to pick the remaining Pallet locations in that aisle.
- In the Shelf area, the alphabetical sorting of pick locations in combination with different location names result in an illogical sequence of pick locations on the pick list. One example where this occurs is in Shelf aisle 7, where the picker first is directed to the front (location 1.3.C), then to the back (location 11.4.A & 13.1.B) and then to the front again (location 3.4.C), as can be seen in Appendix A. This backtracking results in a lot of unneeded travel distance.
- Due to the narrow aisles in the Shelf area, the picking cart cannot be driven into the aisles. Therefore, the order picker has to leave the cart at the entrance of the picking aisle and has to walk into the aisle on his own. In the default route, the cart is first driven to Shelf aisle 11, where the cart is parked so that the picker can visit the pick locations in that aisle. When they return, the cart is reversed carefully to Shelf aisle 7 for the next set of picks. Once finished, the cart is reversed to Shelf aisle 14, which is wide enough for the cart. These manoeuvres, especially the parking and reversing of the cart, are time-consuming.

After completing the route in the default order, a distance of about 210 meters is travelled to visit all locations and retrieve all items.

Route 2 – Improved order

The second route is manually planned to reduce the distance travelled to retrieve all items. Several aspects differ from the default order route:

- Several aisle sections are traversed only once, in which all relevant items are picked during that single pass. For example, Aisle 2 is entered in the beginning of the route and items from both Deck and Pallet locations are picked while the order picker travels through the aisle. This eliminates the need to revisit the aisle after the order picker leaves, as was the case in the default route.
- When leaving Aisle 1, the picker uses the back cross-aisle to reach Aisles 4 and 5. This reduces unnecessary travel and minimises the amount of backtracking required within the aisles.
- In the Shelf area, the picking cart is parked near the entrance of Shelf aisle 7. From there, the order picker walks to Shelf aisle 11 to retrieve the required items from that aisle. After the products are collected, the order picker walks back to the cart, deposits the products and then enters Shelf aisle 7. There, they visit the two upper storage locations, after which they return back to the cart. After reversing the cart into Shelf aisle 14, they walk back to the depot. Before completing the route, the final two pick locations at the bottom of Shelf aisle 7 are visited to retrieve the last items. This approach avoids the need to traverse Shelf aisle 7 multiple times, reducing unproductive travel.

The total length of the improved route is about 150 meters, reducing the travel distance with roughly 60 meters, which is a reduction of about 30% when compared to the default route. Notably, this path is not the result of full route optimisation, but merely a basic reordering and logical grouping of several pick locations. Nonetheless, it shows that by moderately improving the default picking route, a lot of travel distance can be reduced.

Storage assignment

In the warehouse of Company X, different pick locations have different storage assignments. The first two pallet racks in the main picking area mostly contain frequently ordered products of one customer. The reason for this assignment is logical, since the products of this customer are shipped with the same courier, meaning that the orders are batched together. Placing these products close to each other results in less travel distance, thus improving efficiency. However, for the rest of the pallet locations, the products are located somewhat randomly. This can cause some batches to contain products that are stored throughout the whole warehouse, resulting in long picker routes.

In the shelf area, products are also stored on a mostly random basis. Some products are stored close to each other based on their volume, making sure that extremely small items, often stored in boxes, are located near each other. Other products are located randomly throughout the shelves.

Challenges

The current picking process experiences several challenges, due to different aspects. First, the use of paper list results in a lack of verification, as there is no real-time feedback or error prevention. When a picking mistake is made, it is not discovered until the packaging of the products. Furthermore, the paper list only contains the pick locations in alphabetical order, letting the order pickers determine their own routes, which leads to inconsistencies and inefficiencies. There is no policy for a route that the order picker should walk, let alone an optimised route that minimises the total travel time. The paper list also offers no clear guidance for the order picker on where to go next, resulting in excessive walking and backtracking. Sometimes it occurs that a picker is distracted or does not pay enough attention when looking for the next location, resulting in additional search time to look what the correct location is. Lastly, the shelf area has very narrow aisles, which prevents the use of picking carts. This can slow down the picking process in this area.

2.3 KPIs, constraints & requirements

To get a better understanding of the performance and operational challenges at Company X, this section analyses the key performance indicators (KPIs), existing constraints, and process requirements. The KPIs provide insight into how well the warehouse operations are executed, while the constraints show limitation impacting efficiency and scalability. The requirements reflect what is needed to meet customer expectations.

2.3.1 Key performance indicators

Several Key Performance Indicators (KPIs) have been identified and analysed to evaluate the operational performance of Company X. These KPIs focus mainly on efficiency and effectiveness of the picking process, but also on the overall outbound process. For the evaluation of these KPIs, a total of nine batches were used, each with different numbers of picklists and products to get a representative analysis. These batches were randomly selected over different workdays, to ensure a representative and unbiased analysis. In total, 171 picklists were analysed with a combined total of 566 products. Each picklist corresponds to a separate customer order, so 171 different customer orders were analysed.

Lead time

The lead time measures the time between when a customer order is placed and when it is shipped. With this measure, the total response time of the warehouse is reflected, since it shows how quickly a customer order can be fully processed. The lead time is measured per picklist, since each picklist corresponds to one customer order.

As shown in Figure 2.10, lead times vary widely across the analysed batch. Some picklists having a total lead time of only a few hours, whereas a few others have a lead time of more than 168 hours, which equals a full week. The averages of the batches lie between 7 hours and 67 hours.

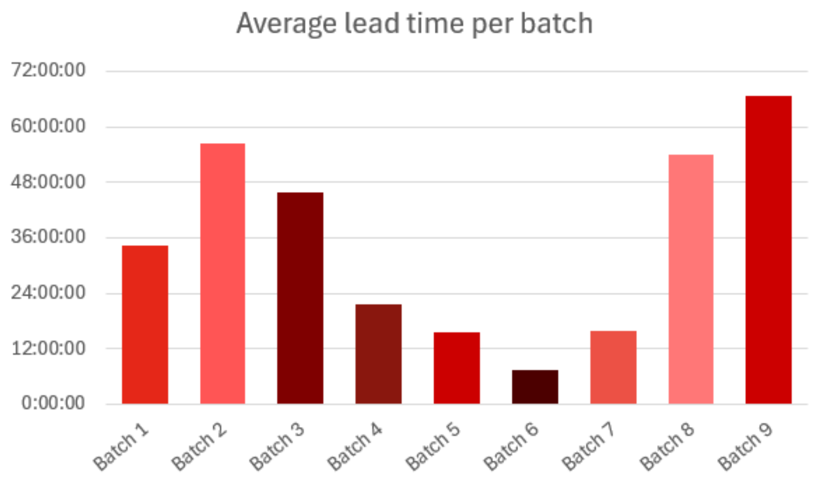


Figure 2.10 Average lead times per batch

Although it seems that most batches have extreme high lead times, it should be noted that these averages are influenced by factors that cause high lead times. Some of these factors are:

- Orders that are placed in the evening cannot be processed the same day. The earliest they could be processed is early in the morning of the next day, which could be already more than 12 hours later. However, if the order is not directly processed, since other orders are prioritised based on their pickup window, the lead time can increase even further.
- Weekend orders suffer even more from this delayed processing, as Company X does not operate during weekends. If, for instance, an order is placed on Friday at 18:00, it will not be processed before Monday 09:00, resulting in a lead time of minimal 63 hours.
- When products are not in stock when they are ordered, they are backordered. Since the order cannot be fulfilled, it needs to wait until a replenishment shipment has arrived and has been processed at the inbound department. Only then the order can be processed and shipped, possibly resulting in extreme high lead times.

Although the lead times of some orders can be high, it should be noted that high lead times not necessarily mean bad customer service. Customers know that when they place an order after working hours, the order cannot be processed until the next day. High lead times only become problematic if the result is that they cannot meet the carrier pickup window. Then, customers cannot get their order in time, resulting in customer dissatisfaction.

Outbound throughput time

The time between the moment a batch is formed and the time an order is ready for shipment is expressed in the outbound throughput time. This measure shows the time it takes to perform all the outbound operations, including picking to packaging, thus reflecting the operational speed.

About 58% of all picklists have a throughput time of less than 2 hours. This means that most orders can be picked, packaged and made ready for shipment in a relatively short amount of time. Often, the picklists that have short throughput times are batched together, as can be seen in Figure 2.11, where the average throughput times per batch are shown.

Out of the analysed batches, five batch averages are below 2 hours, while the other batches have higher average throughput times. Two batches have particularly high averages, one close to 8 hours and one with more than 15 hours. It should be noted however that in Batch 4, there was one particular order that skewed the average significantly due to missing information in the shipping data. Because of this, the supplier had to be contacted to retrieve the correct information, which added significant time. Batch 7 contained some large orders that needed additional processing time before they were picked up, which resulted in extreme high average throughput times.

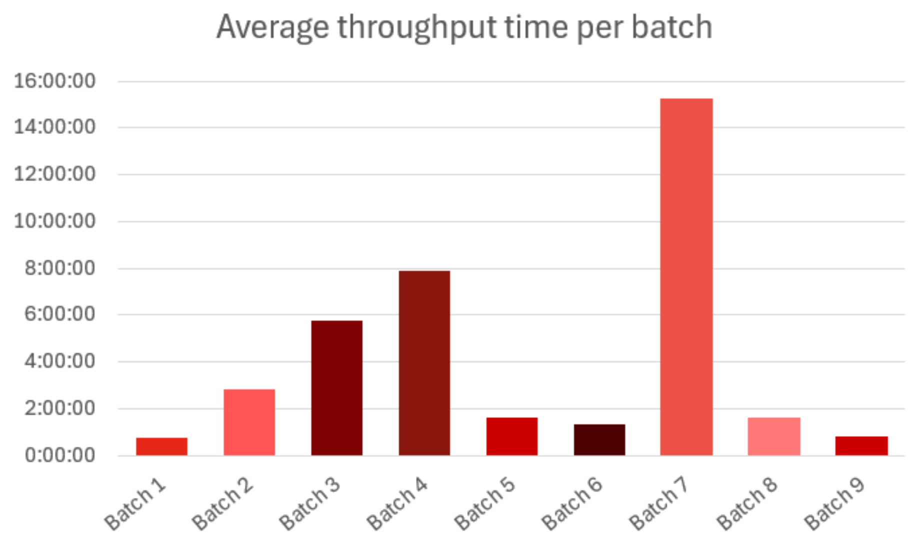


Figure 2.11 Average throughput time per batch

In general, long throughput times are a consequence of a couple of things. First of all, it is possible that many orders come in at the end of a workday. When these orders are picked up at 10:00 in the morning of the next day, it can occur that the batch is created and partly picked. When the workday ends and the batch is not complete yet, it stays open until the next morning to be completed. Since the batch is not finished, the overall throughput time of some picklists become very high.

Another reason for high outbound throughput times can be external data issues, such as missing or incorrect shipping data. This requires the order picker to contact the fulfilment client to retrieve the additional information. Sometimes, this can take some time since the client does not respond immediately, resulting in high throughput times.

Similarly to high lead times, high throughput times are not intrinsically bad. As long as customer expectations are met, it does not matter too much if an order was processed in under an hour or in a couple of hours. While the customer expectation cannot be measured directly, it can be indicated by how often the courier's pickup deadline is met, since this means that the order is picked, packed, and made ready for shipment in time, like the customer expected.

Picking time

The picking time reflects the time that is actually spent on picking of orders in a batch. The total time per batch is measured, so average time per picklist and product can be retrieved. These values are shown in Table 2.2, in addition to other information about the analysed batches. Here, a unique product corresponds to a specific pick location that has to be visited, e.g. Batch 1 has 26 pick locations with a total of 30 products to be picked on these locations.

Table 2.2 Picking times per batch

Batch number	*Number of picklists*	*Number of unique products*	*Total products*	*Picking time*	*Picking time per picklist*	*Picking time per product*
Batch 1	21	26	30	15:29	00:44	00:31
Batch 2	20	23	32	14:51	00:45	00:28
Batch 3	23	23	40	12:53	00:34	00:19
Batch 4	40	29	90	14:19	00:21	00:10
Batch 5	12	7	12	06:16	00:31	00:31
Batch 6	15	22	69	22:34	01:30	00:20
Batch 7	4	30	160	16:49	04:12	00:06
Batch 8	20	17	37	12:13	00:37	00:20
Batch 9	16	34	96	19:14	01:12	00:12
Sum	171	211	566	2:14:38		
Average	19	23.44	62.89	14:58	00:47	00:14

It can be seen that the picking time is not directly related to the number of picklists or products. A comparison between Batch 6 and Batch 9 indicates this:

- Batch 6 contains 15 picklists and a total of 69 products
- Batch 9 contains 16 picklists and a total of 96 products

Given that Batch 9 has both more products and one more picklist, one would expect that it also has a higher picking time. However, Batch 6 took almost 3 and a half minutes longer to complete. The reason for this is that the products in Batch 6 where more scattered around the warehouse, resulting in longer travel times. In Batch 9, the products were grouped more together, which allowed for a more efficient route for the order picker. Additionally, some of the products picked in Batch 6 had high volumes, which required the order picker to reorganise some of these products.

Another comparison should be made between Batch 4 and Batch 7, since they have the same product types, meaning that the products are located in the same storage area. However, the two batches show some noticeable differences in their picking time characteristics due to the nature of the orders per batch.

- Batch 4 contained business-to-customer (B2C) orders, which typically have many small orders consisting of only a few products. This results in a relatively large batch of 40 picklists with a total of 90 products.
- Batch 7, on the other hand, contained business-to-business (B2B) orders, consisting of large, combined orders. The batch only has 4 picklists but a total number of 160 products.

The difference in order structure results in some differences in the picking efficiency. In Batch 4, it took about 10 seconds for a product to be picked, whereas that only took about 6 seconds in Batch 7. This is the case because there was more concentrated picking in Batch 7, since the order picker could collect multiple unis without moving once they were at the correct location. However, when looking at the picking time per picklist, it shows the reverse: Batch 4 outperforms Batch 7 significantly. This is a direct consequence of the order profile, since small orders can be picked and finished quicker.

These examples show that picking times should be evaluated in a given context. Given a number of picking lists and products, the picking time cannot be predicted, since additional aspects have an influence on the time it takes to pick a certain batch.

Travel distance

Travel distance is a performance indicator that directly reflects the efficiency picking routes, since it influences picking time, worker fatigue, and overall productivity. However, in the current warehouse setup, travel distance is not explicitly tracked. To gather the data required to evaluate this KPI, the values were approximated by overlaying the picklist sequence on the scaled layout plan. This method provided a reasonable approximation of the total travel distance per batch. Two types of distances are being evaluated for each batch, namely the ordered route and the followed route. The ordered route follows the picklist sequence exactly as it is listed, whereas the followed route follows the actual route that the order picker took during the picking process. In Table 2.3, the estimated distances and differences between the two types of routes are shown.

Table 2.3 Approximated travel distance per batch

Batch	*Ordered route (m)*	*Followed route (m)*	*Difference*
Batch 1	360	275	31%
Batch 2	380	220	73%
Batch 3	240	195	23%
Batch 4	165	80	106%
Batch 5	155	140	11%
Batch 6	415	380	9%
Batch 7	145	80	81%
Batch 8	325	285	14%
Batch 9	495	320	55%

It can be seen that all followed routes are shorter than the ordered route, showing that the order pickers adjusted the picking sequence to reduce the travel distance. The largest absolute difference between the ordered and followed route is observed in Batch 9, where a total of 175 meters is reduced, equal to a reduction of about 55%. Batch 2 also has a large absolute difference, namely 160 meters, which is a reduction of about 73%. The largest proportional difference is found in Batch 4, where the travel distance in the followed route is more than half the distance of the ordered route.

Smaller reductions are found in Batch 5 and Batch 6, with reductions of 11% and 9% respectively. The reductions of these batches are less significant because both batches have orders scattered all over the warehouse, providing less options for the order picker to walk the picklist in a different sequence. A visual inspection of the pick locations in Batch 6 confirmed it had a scattered batch profile, resulting in a lot of unproductive travel. This also resulted in a high picking time, as discussed earlier.

The analysis of the travel distance shows that there is considerable room for route optimisations. Currently, this optimisation is done by the intuition and knowledge of the order picker, which results in more efficient routes than the ones provided by the WMS. To systematically achieve these improvements, routing optimisation algorithms should be integrated into the system.

Picking accuracy

Currently, no systematic data is available to quantify picking accuracy. Picking errors are not tracked in the WMS, and due to the lack of barcode or scanning validation during the picking process, it is hard to measure how often orders are picked incorrect or incomplete. A picking error is only discovered during the packaging of an order once the packaging employee scans the product to create a shipping label. Once they discover that an item is wrong, they have to replace the wrong item with the correct one, requiring them to walk through the warehouse for only one product. Even though it is hard to measure how often a picking error occurs, it remains an important KPI for evaluating the overall process reliability, especially when order pickers deviate from the provided picklist order, which increases the chance of errors. To increase picking accuracy over time, it is important to implement error tracking mechanisms.

2.3.2 Current limitations and operational needs

To fully understand the warehouse operations at Company X, it is important to acknowledge both the limiting factors of the current situation and the operational needs required to maintain efficiency, scalability and consistency. The following section describes the key constraints that currently influence performance, followed by the fundamental requirements that must be met to remain reliable in the future.

Current limitations

The picking process is directly limited to the number of available employees. Generally, the outbound process is handled by a small team, making it difficult to scale up during peak periods, such as on Monday mornings when accumulated weekend orders are processed. While exact figures on individual picker performance are not tracked, observations indicate that there is variability in the routing efficiency and picking speed between employees, depending highly on their experience and familiarity with the warehouse layout. The reliance on manual labour, in combination with the variable performance of the employees, makes it challenging to maintain consistent throughput times.

Since the picking process is executed primarily manual, human errors can easily happen. Without the use of validation systems, missed picks or picking incorrect items can occur without even being noticed until the final packaging stage. Additionally, a manual picking process does not allow for specified routing, which can help to optimise routes in real time.

The shelving area in the warehouse of Company X is designed with very narrow aisles. The advantages of this are that an order picker can easily pick products from both sides of the walking aisle and that the space utilisation is high. However, due to the restricted access, picking carts cannot be brought into the aisles, which forces pickers to leave the cart at an entrance and walk in the aisle on their own to retrieve the items.

Operational needs

Despite the constraints, Company X must meet some requirements to ensure that they are seen as a reliable, efficient and scalable business partner for their customers. These requirements are influenced by customer expectations and internal performance goals.

The ability to process, package and ship orders within the designated courier pickup windows is one of the most important requirements. To ensure that different couriers can be serviced on time, warehouse operations must be aligned properly, ensuring that correct orders are prioritised.

With increasing outbound volumes, the margin for errors shrinks. The warehouse must achieve consistently high levels of picking and packaging accuracy to ensure that customer expectations are met and return rates are reduced. The risk of human error is increased by the current manual processes, which underscores the need for improved accuracy mechanisms.

Walking times account for a significant portion of the picking process. To minimise these times, optimised picker routes become essential. Currently, the routes are printed on paper, but to ensure that order pickers follow the optimal paths, the information of the next pick location should only become available when the current pick location is finished.

2.4 Conclusion

This chapter analyses the current operational context of Company X to give answer to the research question "How does Company X currently operate?". The analysis focuses on the warehouse layout, important processes, and performance metrics, providing the foundation for understanding the operations that are performed by Company X.

In the warehouse, four key processes are carried out daily. In the inbound process, incoming goods are received and checked. Once accepted, the storage process begins, in which the products are stored in the designated storage areas. When customer orders come in, the outbound process starts. First, orders are picked by order pickers in batches, after which the items are packaged and made ready for shipment. When orders return due to customer dissatisfaction, the return process is started, ensuring that the returned goods are checked and handled properly. A warehouse management system supports all of these processes, although some operations are still carried out manually.

For the evaluation of the performance of the warehouse, different key performance indicators are used, including lead times, picking times and travel distances. The variation found in the metrics indicate that factors such as order profile, routing strategies, and storage allocation influence the performance of the picking process. Some challenges at Company X are related to the number of employees and lack of automation. This leaves room for improvement in areas related to improving picking accuracy and enabling more efficient routings.

3. Theoretical background

In this chapter, information is gathered about different techniques and technologies that can be applied to optimise order picker routing. By doing so, the second research question is answered, which states: *"Based on the literature, what approaches and innovations can be used to optimise or transform order picker routing processes?"*. Section 3.1 introduces the order picker routing problem as a combinatorial optimisation model. In Section 3.2 characteristics of the problem at hand are given. In Section 3.3, algorithmic approaches are explored, distinguished between exact methods, which provide optimal solutions for specific cases, and heuristic & metaheuristic methods, which offer flexible and scalable alternatives. Section 3.4 explores available technologies that are used to support or transform the order picking routing process in warehouses. Finally, Section 3.5 summarises the chapter with concluding remarks.

3.1 Foundational concepts in order picking

A critical and complex aspect of the outbound process of warehouses is the efficient routing of order pickers. With the increasing demand for fast and accurate deliveries, it becomes more essential to optimise picker routes to improve warehouse productivity and reduce operational costs. This section shows how the order picker routing problem can be modelled using a combinatorial optimisation framework such as the travelling salesman problem or the vehicle routing problem.

The picker routing problem is a classic example of a combinatorial optimisation problem. Generally, it can be described as a variation of the travelling salesman problem (TSP), where each pick location is treated as a city that needs to be visited. In more complex situations, it can also be described as the vehicle routing problem (VRP) (Masae et al., 2020). In the VRP, each node must be visited as well, but here, numerous routes can be produced to pass through all the nodes. Figure 3.1 shows the difference between the TSP and VRP, in which the TSP is a single-route node-service combination problem without vehicle capacity limitations and the VRP is a multiple-route node-service-combination problem with vehicle capacity limitations (Liu et al., 2014). In a warehouse context, each item location functions as a 'city' or 'customer', where a picker must traverse a route visiting all required locations before returning at the depot.

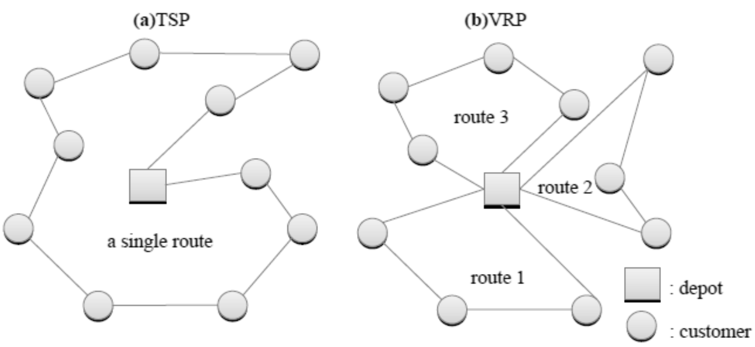


Figure 3.1 Route patterns of (a) the TSP and (b) the VRP (Liu et al., 2014)

Due to specific warehouse layouts like the parallel-aisle structure, the classical TSP cannot be applied for solving the picker routing problem. By exploiting the structured layout of conventional warehouses, Ratliff and Rosenthal (1983) created an algorithm that can solve the picker routing problem for rectangular single-block warehouses optimally and in linear time. They were among the first to show that this specific problem is tractable when properly structured.

In their model, Ratliff and Rosenthal (hereafter referred to as R&R) define a graph representation G of a warehouse, as can be seen in Figure 3.2. This figure shows how a traditional warehouse layout can be mapped to a graph model suitable for the optimisation model of R&R. In the graph representation, vertices represent item locations (v_1, ..., v_{12}), the depot (v_0) and aisle entry/exit points ($a_1 – a_6$ and $b_1 – b_6$, respectively). Travel paths are represented by edges, each edge with a weight corresponding to the travel distance between the vertices connected. A pickers route must visit all required item locations starting and finishing at the depot.

The key concept in R&R's approach is the construction of partial tour subgraphs (PTSs) and equivalence classes, the latter storing the degree of the aisle's endpoints and the connectivity across aisles. Tour subgraphs across aisle sections are constructed iteratively by the algorithm, where the shortest path within each equivalence class that satisfies Eulerian path conditions (i.e., even-degree nodes and connectivity) is selected. R&R showed that there are seven equivalence classes, with only four of them being valid for a final tour subgraph. This results in an overall time complexity that is linear in the number of aisles, making the algorithm practical for real world applications. As they noted themselves, a problem where the warehouse has 50 aisles can be solved in under a minute, where the number of items in the order has little effect on the solution time. Because of the easy to use algorithm and great practical value, a lot of extensions to the model of R&R have been made, which are discussed in Section 3.3.2.

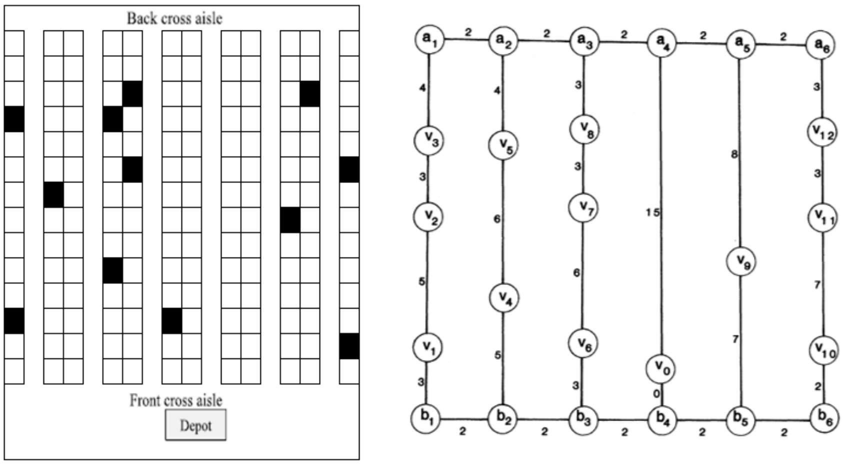


Figure 3.2 Conventional warehouse with a single block (left) and graph representation G (right) (Ratliff & Rosenthal, 1983)

While a variant of the TSP can often be used to model the picker routing problem in warehouses, a broader formulation is required for more complex warehouse systems. For these situations, the VRP can be applied, since it generalises the TSP by allowing multiple routes that serve the demand points together. Often, there are additional constraints in VRPs, such as vehicle capacity limits, time windows or availability of the visiting vehicles. In the warehouse environment, this formulation becomes relevant when there are multiple order pickers that operate with batch or zone picking (Masae et al., 2020). In these situations, a single-route solution becomes less practical or efficient.

In the context of a warehouse, an order picker can be seen as the vehicle and pick locations are treated as city nodes that needs to be visited. The orders to be picked may be split up across multiple pickers, due to workload capacity constraints, aligning with the classical constraints of the VRP (Aerts et al., 2021). As shown in Figure 3.1, in the TSP, a picker services all item locations on their own in one continuous loop, whereas the VRP allows several routes, dividing the workload. This approach increases flexibility of the model, although it also adds computational complexity. The number of feasible route combinations grows rapidly when the number of pickers and order lines increase. Because of this, models to solve the VRP in warehouses often make use of heuristic or metaheuristic techniques, more on this in Section 3.3.2.

While the picker routing problem can be modelled based on the theoretical framework of either the TSP or VRP, a direct implementation into warehouse environments is often not applicable. To make the formulations useful in practice, some adaptations to the models need to be made, leaving a couple of limitations. First, warehouses have different structures, where the layout may differ between traditional parallel aisles or non-conventional layouts. Besides the overall layout, the number of cross aisles, fixed depot locations and possible directional travel rules also deviate from the generic assumptions of the TSP and VRP models. In both TSP and VRP models, paths are unconstrained, meaning that the visitor can directly travel between two nodes. In the picker routing problem, the picker cannot directly travel from one pick location to another but they have to follow aisles (Scholz et al., 2016). Additionally, congestions in (cross) aisles may make them unavailable, resulting in additional complexities. Furthermore, dynamic factors, such as changing priorities or updated orders further complicate routing decisions.

To make the TSP and VRP models applicable for warehouse environments, these constraints must be explicitly integrated. Often, this is done by adapting the distance matrix to the warehouse layout and adding operational constraints like time windows or zone boundaries. However, even with these additional constraints and adaptations, the complexity of the warehouses limits the scalability of exact solutions, making hybrid approaches, where optimisation models are combined with heuristics, an increasingly popular solution (Casella et al., 2023).

3.2 Characteristics of the order picking problem

To understand the algorithmic approaches used for the optimisation of order picker routings, it is important to investigate the characteristics that affect the complexity of routing problems in warehouses. Figure 3.3 presents a state-of-the-art classification of elements that define and influence the order picker routing problem, created by Masae et al. (2020). Their research considers two dimensions of the order picker routing problem, being *Problem characteristics* and *Algorithm characteristics*, where the first dimension consists of both layout characteristics and warehouse operational constraints.

In the warehouse layout, physical aspects of the warehouse are outlined, such as the aisle structure, depot characteristics and rack levels. These aspects determine the environmental basis of the routing problem. In the warehouse operational constraints, issues such as picking strategies, storage assignment and tour planning are addressed. These factors influence the complexity and structure of the picker routing problem. The algorithm characteristics are related to commonly applied methods for solving the routing problem, varying in complexity and scale.

The elements highlighted in red represent the characteristics that are most applicable to the context of Company X, whereas the elements in orange provide additional contextual support, as they are not direct constraints but still need to be considered. The relevant problem characteristics are in Section 3.2.1 (Warehouse layout) and Section 3.2.2 (Warehouse operations).

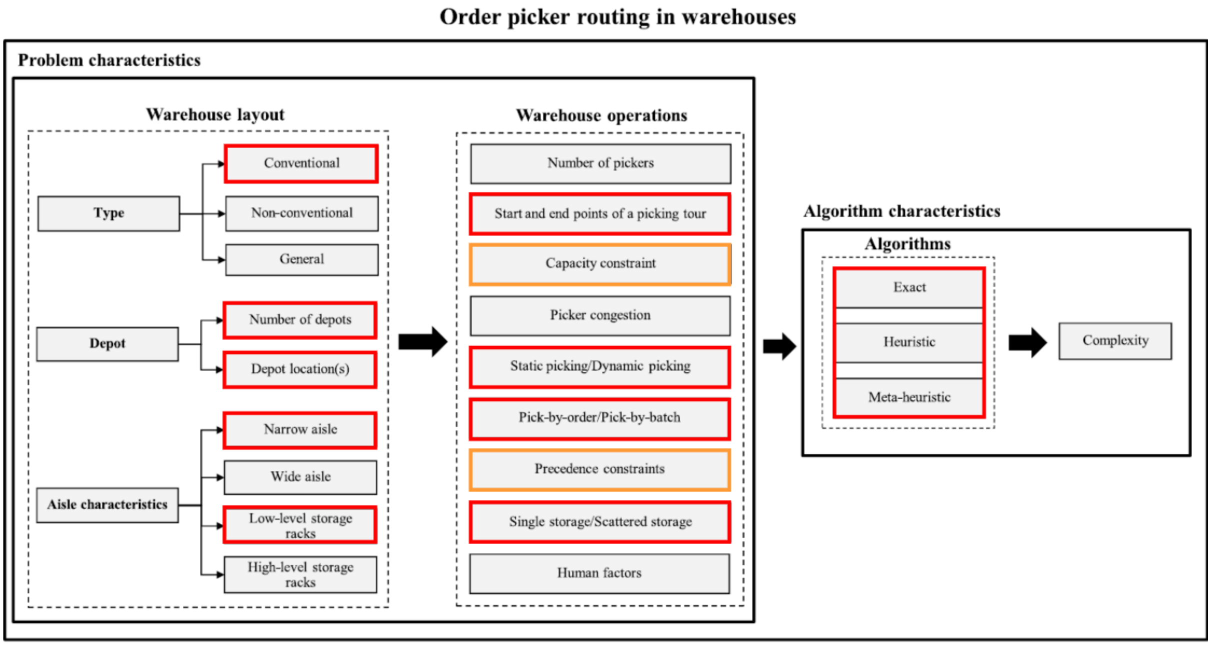


Figure 3.3 Classification of characteristics of the picker routing problem (Masae et al., 2020). Red highlighted items represent the primary focus of this research, while orange highlighted elements provide contextual support

3.2.1 Warehouse layout

The layout of a warehouse relates to the general type, the number and location of the depot(s) and some aisle characteristics. The main warehouse type is the conventional warehouse, which have a rectangular shape with parallel aisles. Perpendicular to these picking aisles are cross aisles that serve to provide access to the picking aisles. In one-block warehouses, there are just two cross aisles on the front and back, whereas multi-block warehouses have more than two cross aisles (Boysen & De Koster, 2025). Non-conventional warehouses do not have parallel picking aisles but select a different layout to allow for picking while improving space utilisation.

The number and location of the depots have significant impact on the layout of a warehouse. Depots are often located centrally in a warehouse, to allow for easy access and reduced travel distances. Decentralised depots can be located anywhere in the warehouse. This is often done to place the depot close to the outbound area, to ease the shipping process (Boysen & De Koster, 2025). Multiple depots influence the routing problem by providing more options on where to start and end the tour.

There are a couple of aisle characteristics that influence the picker routing problem. First, the width of the aisle, where narrow aisles allow the order picker to pick items from both sides of the aisles without having to cross. In wide aisles however, picking from both sides of the aisle makes crossing necessary, which leads to additional travel distance. Second is the level of storage racks that is used in the picking aisles. Low-level storage racks allow for direct picking of items without the need of vertical travel. In high-level storage racks, such vertical movements are needed to reach the items (Masae et al., 2020).

3.2.2 Warehouse operations

Warehouse operations entail various strategies executed or scenarios encountered that influence the routing of an order picker through the warehouse. One of the most influential factors that influences the picker routing problem is the storage assignment policy. This policy outlines how SKUs are stored in a warehouse. A distinction is made between single storage systems, where an item is only stored in a single storage location, and scattered storage systems, where items are stored in multiple storage locations. For the single storage system, the six most common policies are (a) random storage, (b) closest-open-location storage, (c) dedicated storage, (d) full-turnover storage, (e) class based storage, and (f) family grouping (De Koster et al., 2007). All of these policies affect the concentration of picks in certain areas. In warehouse systems with scattered storage, where SKUs can be stored in multiple locations, the routing of orders becomes even more complicated, since decisions must also take the specific location to pick from into account.

Whether a warehouse is operated by a pick-by-order or a pick-by-batch policy also influences the picker routing problem. The pick-by-order policy is the most basic approach of picking orders. Here, an order picker collects all items of a single customer order before starting with the next order (Boysen & De Koster, 2025). This method is easy to implement and manage, since pickers only have to collect the orders of one pick list. On the downside, it results in long travel distances and low productivity, especially when dealing with a high volume of orders. To overcome these negative aspects, the pick-by-batch policy can be implemented. Batching allows the unification of multiple customer orders to one large pick list, which is then jointly processed in an extended picking tour (Boysen & De Koster, 2025). Once the complete batch of orders is collected the picker returns to the depot, after which they can start the next tour with a new batch (Boysen, De Koster, et al., 2019). This increases the pick density and lowers the amount of unproductive travel times, so a more efficient picking process is enabled.

Most warehouses operate with static order picking, where picklists are not allowed to change once the picking process has started. However, in dynamic picking, new orders might come in during the order picking process, resulting in an updated picklist. This means that the picklist and route must be adaptable and require real-time adjustments (Masae et al., 2020). The picker routing process becomes more complex, since routes need to be updated according to the new set of orders, while also making sure that the remaining route is re-optimised without too much delay. Essentially, it can be seen as a picker route that can be started from all positions in the warehouse, with all remaining pick locations as positions that need to be visited.

This relates to the position where a picking tour can start and end. In most cases, a picker starts and ends their route at the same depot, where the picked items are dropped off and made ready for shipment. However, there are also scenarios where order pickers can deposit the picked items in designated areas, located strategically at the front of a cross-aisle. Here, a conveyor belt transports the picked items towards the packaging area, while order pickers receive a new picklist with items that need to be collected (Boysen & De Koster, 2025). Since there are multiple drop off points, it can occur that an order has a different end location than where they started the picking tour. This influences the complexity of the picker routing problem.

Lastly, there are constraints related to both the capacity of the order picker and the precedence of the orders. If a picker uses a picking cart to transport all the items, there is a restriction on the total weight or number of items that they can pick per tour. Precedence constraints are related with the sequence in which items are collected. If, for example, a picklist contains both heavy and light items, it should be taken into consideration that the heavy items are picked first, such that they do not damage the light, fragile items (Masae et al., 2020). However, these problems can partly be tackled by adjusting the storage assignment. By creating different storage areas, items with the same size can be located near each other, resulting in less need to traverse large distances to collect different size products in a specific sequence.

3.3 Algorithmic optimisation approaches

With the order picking problem and its key characteristics defined, this section investigates different algorithmic approaches to solve it. This review is structured in two parts. First, in Section 3.3.1, algorithms for the order batching problem are examined. Various heuristic and metaheuristic methods designed to group customer orders into efficient batches are covered. The focus is on approaches that aim to minimise total travel distance by maximising spatial proximity. In Section 3.3.2, literature on picker routing algorithms is covered. Here, the sub-problem of finding the shortest path to collect all orders within a batch is addressed. Three categories of algorithms are reviewed, all with different advantages and disadvantages. Exact methods guarantee optimality but are computationally expensive, whereas heuristics are intuitive and easy to implement but often do not provide high quality solutions. Metaheuristics are highly flexible and scalable, although globally optimal solutions cannot be guaranteed.

3.3.1 Order batching algorithms

Order batching is an important sub-problem in warehousing aimed at the grouping of individual customer orders into feasible batches. Effective batches can significantly improve warehouse efficiency, as grouping orders with high spatial proximity can largely reduce the total travel distance and thus the overall picking time required to pick all orders. The problem can be solved with several potential objectives in mind, including minimising order tardiness, balancing picker workload, and reducing total travel time. Given that the picker travel time accounts for the majority of operational costs (Bartholdi & Hackman, 2019), this review focuses on the literature where the primary objective is the minimisation of the travel distance, as this directly relates to the picker travel time.

This section reviews four main families of batching algorithms. Table 3.1 provides a summary of the key studies discussed, outlining their core batching strategy and their methodological contributions. The work of De Koster et al. (1999) is particularly relevant, as this article discusses many different potential algorithms that can be used for solving the order batching problem at Company X.

Table 3.1 Summary of relevant literature for the order batching sub-problem

Article	***Algorithm group***	***Batching strategy***	***Key methodological contribution***	***Capacity constraint***
Dukic & Oluic (2007)	Simple, Seed	FCFS, Seed	Compares simple heuristics in a real-world warehouse context	Implicit
De Koster et al. (1999)	Seed, Savings	C&W(i), (ii), (iii)	Analyses performance of different variants of seed and savings algorithms	Item count
Gibson & Sharp (1992)	Seed	Seed	Foundational comparative analysis of different seed-based rules	Item count
Elsayed and Unal (1989)	Seed, Savings	EQUAL	Formed a hybrid between seed and savings algorithm	Item count
Hsu et al. (2005)	Metaheuristic	GA	First use of metaheuristic for the order batching problem	Volume of picking facility
Žulj, Kramer, et al. (2018)	Metaheuristic	ALNS/TS	Developed a hybrid metaheuristic for the order batching problem	Item count
This thesis	Seed, Savings	Seed, C&W	Adaptation of classic heuristics for a mixed-layout warehouse environment	Item and pick location count

3.1.1.1 Simple heuristics

The most straightforward batching methods are simple, rule based heuristics. The most common of these simple batching methods is the First-Come First-Serve approach. This heuristic adds orders to a batch chronologically as they arrive in the system until a capacity limit is reached. When the capacity is reached, a new batch is formed (Dukic & Oluic, 2007). This method is easy to implement and manage, although it does not consider the spatial location of orders and can thus result in inefficient and scattered batches, leading to excessive travel distance and time.

3.1.1.2 Seed algorithm

Seed-based algorithms, or sequential heuristics, offer a smarter approach by considering the spatial proximity of orders. These algorithms add orders one by one to batches by following two steps: First, a seed (or initial) order is selected from all orders not yet added to a batch. Then, orders that have not been selected are added to the seed order until the batch capacity is reached (De Koster et al., 1999). The seed order can be selected based on:

- its location in the list of available orders; either the first or last added order is selected.
- a random probability, by selecting a random order (Gibson & Sharp, 1992).
- the travel time from the depot, by selecting the order with pick locations furthest away (Dukic & Oluic, 2007).
- the number of items in an order, by selecting the order with the most items.

To determine what order has to be added to the seed order, different seed order addition rules exist, for example:

- Add the order that provides the largest reduction in travel distance when combined with the existing batch (Gibson & Sharp, 1992).
- Add order that minimises the number of *additional aisles* compared to the seed order.

Seed algorithms are often used in practice because of their flexibility and simplicity.

3.1.1.3 Savings algorithm

A more sophisticated approach to form batches is by means of a savings algorithm. These algorithms, often based on the algorithm of Clarke and Wright (1964) for the vehicle routing problem, take a more global perspective than sequential seed-based methods. Instead of building batches by adding orders one by one, these algorithms start with each order in its own batch and then merge them iteratively (De Koster et al., 1999). First, the savings of every possible pair of orders are calculated, where the savings are defined as the reduction in travel distance achieved by combining two orders into one batch, compared to when both orders should be picked separately. Then, the algorithm merges the pair with the highest savings, given that the capacity constraint is met, after which the process starts over again until no more positive savings can be found. De Koster et al. (1999) discussed multiple variants of the C&W savings algorithm, where there is one basic variant, one where the savings are recalculated after every addition of an order and one where there is a limitation on the number of batches.

Elsayed and Unal (1989) created a combination between the seed and savings algorithms. Their algorithm, called EQUAL, selects a combination of two orders as its seed. This combination is chosen based on the largest time saving, which is computed with the savings calculation. The order with the largest potential time saving is added to the batch until the batch capacity is reached. Then, a new pair of orders is selected for the seed, after which the procedure repeats.

3.1.1.4 Metaheuristic

For larger or more complex situations, metaheuristics can be used to find higher-quality batches. These methods use advanced search strategies, which allows them to explore a vast number of possible batch combinations. Hsu et al. (2005) were among the first that applied a metaheuristic approach for the batching problem. By using a Genetic Algorithm (GA), they utilised principles of evolution like crossover and mutation to generate and improve sets of batches. Unlike methods like the seed or savings algorithm, their approach does not require the computation of order or batch proximity, as it directly minimises the total travel distance.

A more advanced approach to solve the order batching problem is developed by Žulj, Kramer et al. (2018). They propose an ALNS/TS hybrid metaheuristic, which they apply purely to the batching decision. The Adapative Large Neighbourhood Search (ALNS) is used for the diversification. By removing a set of orders from a batch (destroying) and then reinserting them into the best possible new batches (repairing), new solutions are created. The algorithm adaptively learns which destroy and repair operators are the most effective and chooses them accordingly. The Tabu Search (TS) component provides the intensification. By performing a local search, it explores neighbouring solutions and moves to the best one it finds, even if it is worse than the current solution. By maintaining a ‘tabu list’, the algorithm prevents the search from immediately reversing, which helps to navigate out of local optima.

Both Hsu et al. (2005) and Žulj, Kramer, et al. (2018) found that their metaheuristic approaches outperformed the classic heuristics like FCFS and seed and savings algorithms. This shows that, especially for large and complex problem instances, advanced metaheuristics are the best choice for solving the pure order batching problem. However, since this thesis does not purely focus on the batching problem alone, it is decided that the most suitable algorithms are the C&W savings algorithm and seed-based Greedy algorithm. These algorithms still provide a strong and efficient foundation for the subsequent routing stage, while the solutions are created more quickly than with a metaheuristic approach.

3.3.2 Picker routing algorithms

Once orders have been effectively grouped into batches, the second sub-problem, the picker routing problem, can be solved. This sub-problem involves finding the shortest possible tour for a picker to visit all required pick locations per batch. The efficiency of the routing algorithm has a direct impact on the total travel distance. Three main classes of algorithms are reviewed: exact methods, heuristics, and metaheuristics.

Table 3.2 gives an overview of the literature that has been used to examine the exact, heuristic and metaheuristic methods. Each article is classified according to the characteristics of the picker routing problem as defined by Masae et al. (2020).

Table 3.2 Summary of relevant literature addressing the picker routing sub-problem

Article	Based on	Warehouse layout						Warehouse operations									Algorithm
		Conventional	Single block	Number of depots	Location of depot	Aisle width	Level of storage	Number of pickers	Same start and end	Capacity constraints	Picker congestion	Static picking	Batching	Precedence constraints	Single storage	Human factors	
Ratliff & Rosenthal (1983)	TSP	☑	☑	1	Central	Narrow	Low	1	☑	☐	☐	☑	☐	☐	☑	☐	Exact
De Koster and Van Der Poort (1998)	TSP, R&R	☑	☑	1	Decentralised	Narrow and wide	High and low	1	☐	☐	☐	☑	☐	☐	☑	☐	Exact
Žulj, Glock, et al. (2018)	TSP, R&R	☑	☑	1	Decentralised	Narrow	Low	1	☑	☐	☐	☑	☐	☑	☑	☐	Exact
Lu et al. (2016)	TSP, R&R	☑	☑	1	Central	Narrow	Low	1	☑	☐	☐	☐	☐	☐	☑	☐	Exact
Roodbergen and De Koster (2001)	TSP, R&R	☑	☐	1	Central	Narrow	Low	1	☑	☐	☐	☑	☐	☐	☑	☐	Exact
Pansart et al. (2018)	TSP, R&R	☑	☐	1	Central	Narrow	Low	1	☑	☐	☐	☑	☐	☐	☑	☐	Exact
Chabot et al. (2016)	VRP	☑	☑	1	Central	Wide	Low	1	☑	☑	☐	☑	☐	☑	☑	☐	Exact
Shetty et al. (2020)	VRP	☑	☐	1	Decentralised	Narrow	Low	1	☑	☑	☐	☑	☐	☐	☑	☐	Exact
Theys et al. (2010)	TSP	☑	☑	1	Either	Narrow	Low	1	☑	☐	☐	☑	☐	☐	☑	☐	Improvement heuristic
Tsai et al. (2008)	TSP	☑	☑	1	Decentralised	Narrow	Low	1	☑	☑	☐	☑	☑	☐	☑	☐	Metaheuristic (GA)
Schrotenboer et al. (2017)	TSP	☑	☑	1	Central	Narrow	Low	Multiple	☑	☑	☑	☑	☐	☐	☑	☑	Metaheuristic (HGA)
F. Chen et al. (2013)	TSP	☑	☐	1	Decentralised	Narrow	Low	2	☑	☐	☑	☑	☐	☐	☑	☐	Metaheuristic (ACO)
Nathania and Komarudin (2023)	TSP	☑	☐	1	Decentralised	Narrow	Low	1	☑	☑	☐	☑	☑	☐	☑	☐	Metaheuristic (SA)
Kulak et al. (2011)	TSP	☑	☐	1	Either	Narrow	Low	1	☑	☑	☐	☑	☑	☐	☑	☐	Metaheuristic (TS)
This thesis	TSP	☑	☐	1	Decentralised	Ultra narrow	Low	1	☑	☐	☐	☑	☑	☐	☑	☐	Metaheuristic (SA)

While Table 3.2 provides an extensive overview of algorithmic approach for solving the picker routing problem, three studies stand out as particularly relevant to this research. This is based on the characteristics of the order picker problem corresponding between the examined studies and this research. These studies are from Lu et al. (2016), Nathania and Komarudin (2023), and Kulak et al. (2011).

The studies of both Nathania and Komarudin (2023) and Kulak et al. (2011) are considered particularly relevant, since they are designed for large, multi-block warehouses, while also incorporating the batching problem. Although the studies use different metaheuristics, simulated annealing and tabu search, they demonstrate that those methods can effectively solve joint decision problems in warehouse environments. For this research, simulated annealing is selected as most appropriate due to its balance between complexity, adaptability, and solution quality. However, the principles from tabu search provide valuable conceptual information.

The following section examines the different algorithms approaches for solving the order picker routing problem. Particular emphasis is given to the three studies highlighted above due to their relevance to this research.

3.3.2.1 Exact algorithms

Exact algorithms aim to solve the picker routing problem by finding the optimal route that minimises travel time or distance. Due to the complexity of warehouse systems, these exact methods are best suited to small or medium sized problems.

For picker route problems based on the TSP, Ratliff and Rosenthal (1983) created one of the most influential works in the field of exact algorithms. This algorithm is tailored to rectangular single-block warehouses with narrow aisles and a single depot in the front cross aisle. Due to the lack of flexibility in different order picking scenarios, numerous extensions have been made to the algorithm. For example, De Koster and Van Der Poort (1998) adapted the algorithm such that it could work with decentralised depositing. Here, the order picker drops items off at the front of an aisle, from where items are transferred to further processing stations.

Further extensions to the R&R model have incorporated real-world constraints. For instance, Žulj, Glock, et al. (2018) proposed a two-phase routing structure that handles precedence constraints, allowing heavy items to be picked before lighter ones. For environments with dynamic picking, Lu et al. (2016) incorporated the interventionist routing algorithm (IRA), which can update a route in real time when new orders arrive.

For more complex layouts, additional extensions have been made to the R&R model. Roodbergen and De Koster (2001) adapted it for two-block warehouses, where one middle cross aisle is added and Pansart et al. (2018) extended it even further by making the model applicable for warehouses with any number of cross aisles.

Lastly, there are some exact algorithms based on the VRP. Chabot et al. (2016) used a branch-and-cut algorithm to solve the picker routing problem in a single block warehouse, while incorporating precedence constraints. Shetty et al. (2020) uses the Miller-Tucker-Zemlin (MTZ) formulations to define the picker routing problem as a capacitated VRP. This way, the problem structure is simplified, while key constraints are still met, resulting in optimal routes for multi-block warehouses.

While exact algorithms result in optimised picker routes, the scalability is often limited due to the computational complexity of the models, especially in large-scale warehouses with dynamic settings. These situations often require the use of heuristic or metaheuristic methods, which are discussed in the next sections.

3.3.2.2 Heuristic methods

Heuristic methods are rule-based strategies designed to quickly generate a good solution. A distinction is made between simple heuristics and improvement heuristics. As the name suggests, simple heuristics are easy to implement and understand. The created routes are often intuitive and it is possible that the routes reduce congestions. Improvement heuristics are used to change an initial solution in such a way that an improved route is formed.

In Figure 3.4, some simple heuristic methods are shown. The most intuitive heuristics include the S-shape, return, and midpoint. By altering or combining some of these heuristics, the largest gap or composite heuristics are created (Roodbergen, 2001). While these heuristics are fast, they can be far from optimal. Therefore, improvement heuristics, such as 2-opt and 3-opt, are often used to improve the route. By iteratively reversing segments, routes with reduced travel distances are created. Theys et al. (2010) showed that applying 2-opt to routes generated by simple heuristics could significantly improve the quality of the routes.

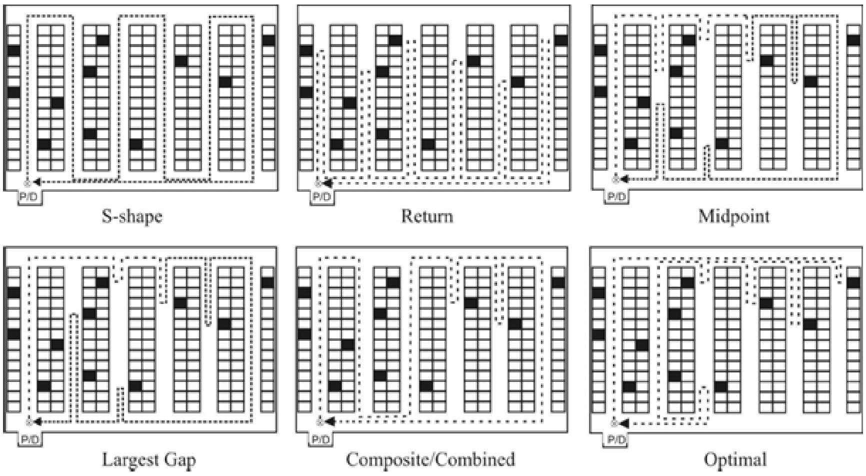


Figure 3.4 Examples of a number of routing methods for a single-block warehouse (Roodbergen, 2001)

3.3.2.3 Metaheuristic methods

Metaheuristic methods extend further than the fixed rules of heuristics by using adaptive search strategies to explore large solution spaces in an efficient manner. By doing so, they can escape local optima to find near-optimal solutions in reasonable timeframes. Metaheuristics are flexible and can be used in large warehouse environments where multiple constraints must be met. In the rest of this section, different types of metaheuristics are discussed.

One metaheuristic that is often used is the Genetic Algorithm (GA). Tsai et al. (2008) proposes two GA methods to solve the order batching and routing problems. The algorithms apply crossovers and mutations to explore different solutions, choosing the best ones based on overall travel distance and due time penalties. Schrotenboer et al. (2017) uses GA to address a warehouse scenario in which delivery of returns is integrated into order picker routes. New possible solutions are created by applying warehouse-specific mutation and crossover operators. To avoid early convergence, infeasible solutions are allowed which are penalised on their fitness value.

Chen et al. (2013) propose a routing algorithm based on the metaheuristic Ant Colony Optimization (ACO) that takes picker congestion into consideration. The ACO metaheuristics uses artificial ants that iteratively construct feasible picking routes based on a pheromone trail and a visibility function. To improve the quality of found solutions, the pheromone trails are updated through evaporation and reinforcement of frequently used paths.

Nathania and Komarudin (2023) present a two-staged simulated annealing algorithm to solve two order picking problems jointly, namely order batching and picker routing. In the first stage, orders are grouped together for the batching decision to provide an initial solution for the algorithm. This is done by a simple integer programming model based on trolley capacity constraints. In the second stage, simulated annealing is used to solve the routing problem for each batch by applying local search heuristics. Simulated annealing algorithms occasionally accept worse solutions, which allows it to escape local optima and explore a broad range of solutions. The change of accepting a worse solution is governed by a temperature parameter, which reduces with each iteration. Their experiments show that this approach found high-quality solutions in only a fraction of the computational time required by an exact branch-and-bound algorithm. This makes simulated annealing highly effective for large-scale warehouses.

Kulak et al. (2011) presents a tabu search (TS)-based metaheuristic approach that is used to solve the picker routing problem in combination with order batching. In this approach, batches are made by grouping orders together, after which feasible routes are created for each batch. By swapping or moving orders between batches, the algorithm iteratively explores neighbouring solutions, whereafter the corresponding routes are recalculated. To avoid cycling or getting stuck in local optima, recent moves are recorded in a tabu list that prevents them from being directly reversed.

3.4 Technological support and innovation in order picking

As warehouses keep evolving due to growing customer demands and tighter fulfilment deadlines, technological advancements are becoming more important in optimising warehouse operations. Especially for order picking, a time and money consuming process, there is an increasing number of supporting technologies. The following section explores some of these technologies, starting with established systems to support order picking, after which emerging innovations with the potential to redefine the picking process are examined. In Table 3.3, an overview of the discussed technologies is shown, in addition to how the technologies can impact different operational KPIs.

Table 3.3 Overview of technologies and their influence on order picking KPIs

Technology	***Efficiency KPIs***	***Accuracy KPIs***	***Adaptability KPIs***
Barcode scanner	Picking speed	Error rate	
RFID	Picking speed	Error rate, Inventory accuracy	Real-time visibility
Pick-to-light	Travel distance, Picking speed	Error rate	-
Pick-to-voice	Picking speed	Error rate	-
Pick-to-vision	Picking speed	Error rate	Real-time guidance
Mobile robots	Travel distance, Picking speed	-	System flexibility, Autonomous coordination
Parts-to-picker	Travel distance, Picking speed	Error rate	System scalability
Intralogistics transport system	Travel distance	-	System integration
AI	Travel distance, Picking speed, Resource utilisation	Error rate, Predictive accuracy	Dynamic decision making, System adaptability
Internet of Things	Travel distance, Picking speed, Resource tracking	Error rate, Inventory accuracy	Real-time visibility, Predictive maintenance

3.4.1 Order picking support systems

As stated before, order picking is one of the most labour- and cost intensive processes in warehouse operations (Bartholdi & Hackman, 2019). Of this order picking time, about 55% of the time is spent on traveling through the warehouse. This means that the other 45% is spent on searching for the right products and being busy with the paperwork of each order. Paperless order picking systems can help reduce the time it takes to fulfil an order, in addition to enhancing accuracy by reducing potential errors (Battini et al., 2015). Some of the most used paperless order picking support systems are described below.

Barcode scanners

Barcode scanners are one of the most widely used technologies in warehouse environments, due to their ability to streamline inventory tracking and picking operations. They come in many different forms, including handheld and wearable barcode scanners (Richards, 2011). By using the scanners, pickers can scan barcodes on storage locations and on products to confirm they pick the correct items. The real-time data transmission to the warehouse management system (WMS), which allows for immediate feedback on errors, significantly improving picking accuracy (Battini et al., 2015). Wearable, finger-mounted scanners have a main advantage over handheld scanners by allowing the picker to operate both hands while handling products. This further improves efficiency, picking accuracy and ergonomics (Stević et al., 2022).

Radio frequency identification technology

Radio frequency identification (RFID) offers a more advanced alternative to barcodes. By using radio waves, items can be identified without the need of a direct line of sight with the scanner. This allows for the simultaneous reading of multiple items, which makes it a highly efficient alternative for barcodes in inventory management and order picking applications (Battini et al., 2015). While RFID can significantly improve accuracy and process speed, the widespread implementation has been limited by the high implementation costs in comparison to barcode scanners and potential signal errors near metals and liquids (Hompel & Schmidt, 2007).

Pick-to-light

Pick-to-light systems guide pickers directly to the correct pick locations using light indicators mounted on shelves or racks. When a pick tour starts, the system illuminates the light at all required pick locations for the specific order or batch. A digital display at each locations shows the quantity that must be picked. Once this is done, the picker presses a button to confirm the action, after which the lights turn off (Battini et al., 2015). This method is highly effective at minimising the search time. Pick-to-light is particularly effective in high-density picking environments with zone picking, as the order picker can see the locations in their designated area instantly (Richards, 2011).

Pick-to-voice

Pick-to-voice systems give pickers verbal instructions through a headset, which is connected to the WMS (Battini et al., 2015). The system directs the picker to a pick location and states the required quantity that must be picked. The picker confirms the location and quantity by speaking in the headset, after which the system provides the picker with the next pick location. The main advantage of the pick-to-voice system is that order pickers can operate both hands-free and eyes-free, improving ergonomics, safety, and allowing the picker to focus purely on the physical task of handling the products.

Pick-by-vision

Pick-by-vision is a relatively new technology that uses augmented reality (AR) via a head mounted display (HMD) or smart glasses. The WMS shows information, such as storage location and picking quantity, directly into the picker's field of view (Haase & Beimborn, 2017). Integrated cameras allow for easy scanning of barcodes, providing instant visual feedback for the order picker. While the technology allows the order picker to operate hands-free and provide visual aids, the widespread implementation is limited by high investment costs, ergonomic challenges, and safety concerns related to partially obstructed vision (Stević et al., 2022).

3.4.2 Emerging innovations

While the picking support systems described above improve accuracy and reduce travel or handling inefficiencies, there are emerging technologies that promise to fully transform the order picking process. These will move a warehouse towards smarter, more autonomous, and data-driven operations.

Mobile robots

Mobile robots, like Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs), are transforming the picking process by automating the transportation of goods. In common applications, these robots meet an order picker in an aisle, who places retrieved items onto the robot. Once the order or batch is completed, the robot travels to the depot independently, while the picker remains in the picking area to meet the next available robot. This significantly reduces unproductive travel time for the order picker, who does not have to travel to and from the depot (Richards, 2011).

Unlike AGVs, which can only follow predefined paths, AMRs can move dynamically, allowing the later to quickly adapt when the operating environment changes. Moreover, AMRs can communicate independently with other resources like control systems and machines, whereas AGVs are controlled through a central unit that makes decisions on e.g. dispatching and routing (Fragapane et al., 2021).

Besides AGVs and AMRs, drones have been used in warehouses as an emerging complement to the ground-based robots. While they are often not used for the actual picking of products due to payload and safety concerns, drones are used for aerial inventory tracking and item location verification (Malang et al., 2023). Tasks that would otherwise require ladders, lifts, or human interaction can easily be performed by drones due to their ability to fly over storage units. By using barcode scanners, drones are able to scan items that would not be visible from the ground-level, resulting in quick, real-time inventory audits. Drones can get a more direct role in the picking process when payload capacity and safety features improve (Kapoor et al., 2024).

Parts-to-picker systems

Parts-to-picker systems reverse the traditional picker-to-parts model by delivering items to stationary pickers by using shuttles, conveyor belts or robotic arms. The main aim is to reduce unproductive travel for pickers. By using Automated Storage and Retrieval (AS/R) systems, shuttle-based systems or AMRs, products are brought to workstations where pickers collect the correct items and process them further (Boysen & De Koster, 2025). After the correct number of items are picked, the system returns the remaining items back to the storage area, where they will be stored until they are required again. This process allows for high space utilisation, since the picking area does not need aisle for human order pickers. Although the initial investment costs are high, parts-to-picker systems eliminate the need for pickers and improves efficiency, making them particularly attractive for environments with high-volume or high-frequency orders (Boysen & De Koster, 2025).

Intralogistics transport systems

Besides optimising the picking task, efficiency can be gained by automating how picked items are transported and sorted. While mobile robots, like the AGV and AMR, handle the transport of entire batches, there are intralogistics transport systems like conveyor and pouch sorters that focus on moving and ordering individual items.

Automated sorting systems use a network of conveyor belts to move picked items from different zones to a central area, where the items are sorted. By using technologies like tilt trays or cross-belts, items are automatically sorted into different channels based on specific customer orders or delivery routes (Boysen, Briskorn, et al., 2019). This system offers high throughput and reliability, in addition to the ability to handle heavy or bulky products.

Pouch sorter systems are a modern and flexible alternative, where each picked item is placed into a separate bag that hangs from an overhead rail (Boysen et al., 2023). These bags can be moved, temporarily stored, and sequenced dynamically, allowing for quick order consolidation. This makes pouch sorters highly valuable for environments where flexible responses to quickly changing demands are required.

Artificial intelligence

In recent years, Artificial intelligence (AI) is increasingly used as a decision making tool to optimise warehouse operations, by analysing data to improve various processes. In the context of order picking, AI can make smart decisions on batching and sequencing by analysing historical picking patterns, order profiles, and warehouse conditions (Nishar, 2024). Moreover, AI can be used for improving other technologies. For example, when AI is integrated into AMRs, pathfinding algorithms enable the robots to navigate dynamically around obstacles and congestion, while it also creates maps of their surroundings in real-time (Sodiya et al., 2024).

Internet of Things

The Internet of Things (IoT) has gained widespread acceptance and popularity in warehouses due to the usage of smart devices and high-speed networks. With this technology, it is possible for devices, pickers, and warehouse systems to seamlessly exchange data with each other (Kumar et al., 2022). By using IoT, the real-time visibility and traceability of products and processes can be improved. For example, products can be detected by smart shelves when they are picked, after which the inventory levels are updated automatically. With the available data, better operational decisions can be made. Sensors mounted into equipment enables tracking of safe behaviour by users, in addition to scheduling of maintenance (De Koster, 2022).

3.5 Conclusion

In this chapter, the theoretical background for solving the order picking problem at Company X is reviewed. The problem is first formulated as a combinatorial optimisation problem, grounded in commonly used models like the TSP and VRP, although these models are not directly applicable due to the unique constraints of a warehouse environment. The specific characteristics of the problem at Company X, most notably the joint batching and routing problem and the mixed layout of narrow and ultra narrow aisles, are formally defined to provide a clear problem context.

The review of algorithmic approaches is structured in two parts to find methods for solving the order batching problem and the picker routing problem, respectively. For the first problem, different heuristics families such as seed-based and savings algorithms are shown to provide efficient and practical methods of grouping customer orders together based on their spatial proximity. For the picker routing problem, a wide range of methods are examined. Exact algorithms can produce optimal solutions for small, structured warehouses, although they are not scalable to larger, more dynamic environments. Heuristic methods are less complex due to their fixed decision rules, although this results in lower quality solutions. Metaheuristic algorithms can offer near optimal solutions within good timeframes, while they also offer adaptability and scalability. Especially the works related to simulated annealing and tabu search are found to be highly applicable to this research, given their practical balance between solution quality and computational efficiency, in addition to their ability to handle joint decision problems like order batching and picker routing.

In the final section, technological innovations that support order picking are examined. Support systems such as barcode scanners and pick-to-voice systems aim to improve the picking process, with improvements in both picking speed and accuracy. More advanced innovations are used to transform the picking process from a manual to more automated process, where mobile robots and smart conveyor belts are used to transport items throughout the warehouse, reducing the travel distance of order pickers. Technologies such as AI and IoT can transform warehouses to become even more automated, improving real-time tracking of inventory and other resources.

This comprehensive review reveals two significant gaps in the existing literature. First, there is limited research on the optimisation of the picker route in warehouses that have a mixed layout with both standard and ultra-narrow aisles. Second, most research treat the streams of algorithmic optimisation and technological support separately.

This thesis aims to address these gaps by developing a tailored optimisation model for Company X's unique mixed aisle environment. Subsequently, in the experimental phase, the performance of the framework will be evaluated under various technological scenarios, providing insights into the synergies between advanced batching and routing algorithms and supportive technologies.

The next chapter introduces the design of the solution approach by first decomposing the problem into two stages, the batch formation and the route optimisation. Then, the problem is formally defined as a Two-Stage Order Picking Problem with Batching and mixed-layout Ultra-Narrow Aisles (TS-OPPB-UNA), after which the proposed algorithms to solve this problem are provided.

4. Solution approach

This chapter presents the development of the solution approach that optimises the order picking process at Company X. Hereby, it gives answer to the third research question: *"How can a solution approach be designed to improve the efficiency of Company X's order picking process?"*. First, in Section 4.1, the problem of Company X is decomposed into two stages, after which the formal definition of the problem is given. In Section 4.2, the first stage, batch formation, is discussed, where customer orders are grouped efficiently together. Section 4.3 discusses the second stage, route optimisation, where the routes for each batch are determined.

4.1 Problem definition

This section formally defines the operational challenge at Company X as a joint batching and routing problem. First, the overall problem context is established and the decomposition of the problem into two manageable stages is explained. Then, the problem is formally defined, which includes key requirements and assumptions.

4.1.1 Problem context and decomposition

The order picking process is one of the most time consuming activities in warehouse logistics, resulting in significant impacts on operational efficiency and costs. The objective of Company X is to streamline this process by optimising the route taken by order pickers while ensuring that all customer orders are fulfilled accurately and on time. This optimisation problem can be described as a joint batching and routing problem, which involves two tightly related decisions:

1. Order batching: How should individual customer orders be grouped together into batches for a single picking tour to ensure maximum pick density and minimum unproductive travel?
2. Picker routing: For each batch created, what is the shortest possible route for an order picker between all required pick locations, starting and ending at the depot?

Solving both of these problems simultaneously is known to be NP-hard and for the instances of the size and complexity at Company X, this becomes computationally intractable. A solution approach that focuses on optimising both the batch composition and the route creation at the same time would become impractical for the real-time operational needs of Company X. To ensure that the solution is practical and effective, the joint problem is split into a sequential, two-stage optimisation framework. This approach focusses on each sub-problem systematically, proving a balance between the solution quality and the computational feasibility. The framework, shown graphically in Figure 4.1, contains the following stages:

1. Batch formation: In this stage, all customer orders are grouped into feasible batches. The main goal is to form densely packed batches by grouping orders based on their spatial proximity.
2. Route optimisation: In this stage, each batch formed in Stage 1 is addressed separately. For each batch, an optimal or near optimal picking route is created that minimises the total travel distance required to pick all items.

This decomposition allows for the use specialised algorithms for both stages, where the output of the batch formation stage is directly used as input for the route optimisation stage, creating a logical workflow. In Figure 4.1, these specialised algorithms are noted, although they will be discussed more in depth later in this chapter. It is important to note that for the route optimisation stage, a metaheuristic approach is used to find near-optimal picking routes, while a separate mathematical model serves as a benchmark to validate the quality of the metaheuristic.

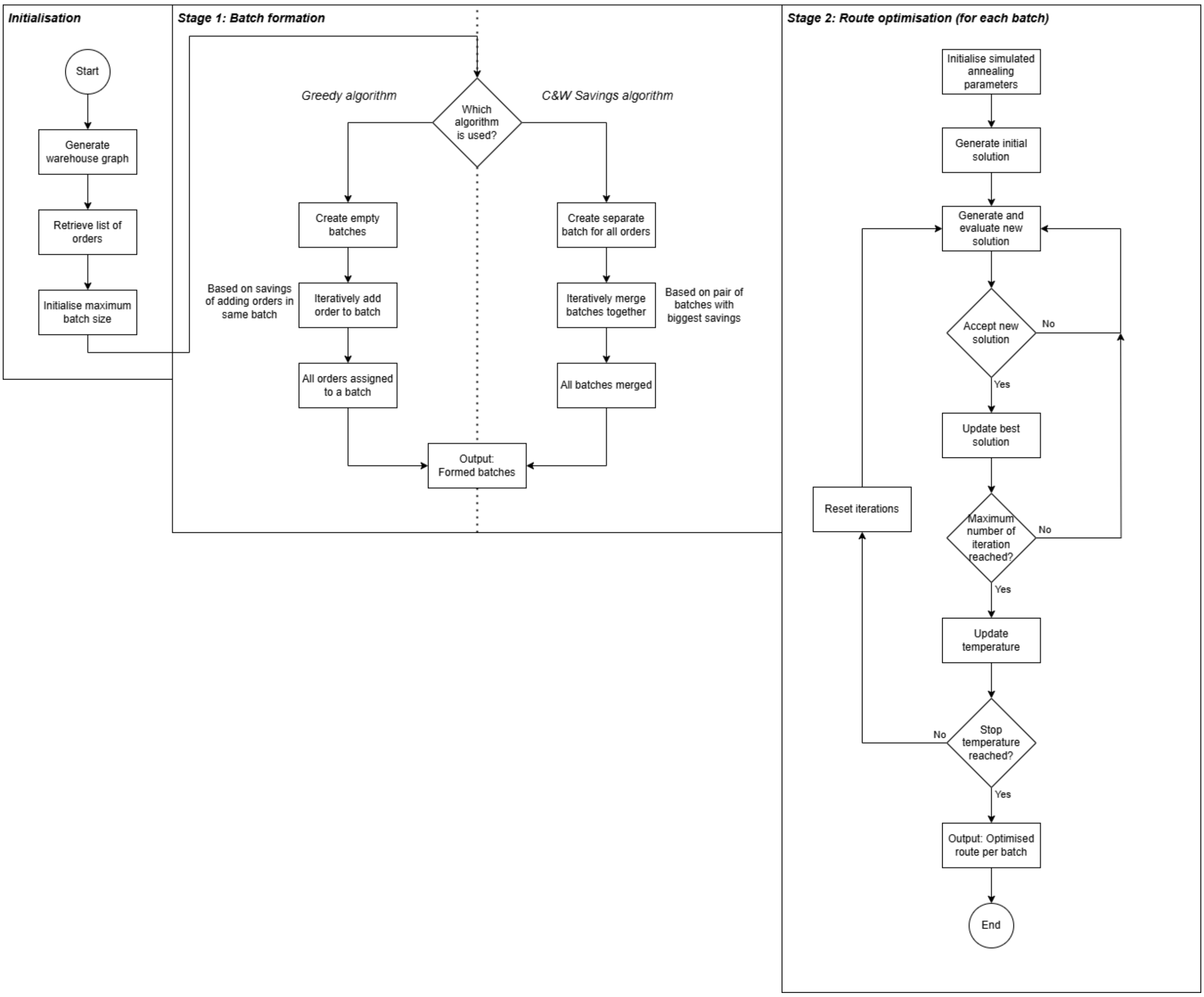


Figure 4.1 Flowchart of two-stage optimisation framework

4.1.2 Formal definition and taxonomy

The order picking problem at Company X can be formally described by defining the two main components of the problem: the physical warehouse environment and the set of customer orders that need to be fulfilled. Besides the formal problem description, this section proposes a taxonomy to classify the problem at hand.

Warehouse environment

The warehouse environment is represented as an undirected graph $G = (N, A)$, where the set of nodes N represents all relevant positions in the warehouse. The nodes set N consists of three distinct subsets, being:

- L, the set of all storage locations
- W, the set of walking points
- $\{d\}$, the depot node

Combined, this makes that the node set $N = L \cup W \cup \{d\}$. The set of arcs is represented by $A \subseteq N \times N$, where an arc is defined as (i, j) with $i, j \in N$. The set of pick locations $P \subseteq L$ contains all the locations with items that need to be picked in a tour. The set R represents all the ultra-narrow aisles, with $N_r \subseteq N$ the set of all nodes within ultra-narrow aisles, where $r \in R$.

The warehouse environment is subject to several fundamental structural assumptions which ensure that the model represents the real world as closely as possible:

- **Fixed storage locations**
  It is assumed that each item has a single and fixed storage location within the warehouse that does not change over time.
- **Deterministic travel distance**
  The travel distances between nodes are assumed to be deterministic and known in advance, implying that it does not vary from operational uncertainty or human behaviour.
- **No congestion**
  It is assumed that order pickers can move freely throughout the warehouse without encountering congestions or obstructions in aisles caused by other personnel or equipment.
- **Perfect information availability**
  It is assumed that complete knowledge of all orders, item locations, and warehouse layout is available at the time of the route creation, leaving no uncertainty regarding stock availability.
- **Picker cart movement**
  It is assumed that the picker cart can move around the warehouse without any movement constraints, except for the shelving area.
- **Mixed aisle structure:** The warehouse consists of two primary aisle types:
  - **Narrow aisles:** In the main pallet rack area, aisles are considered narrow, meaning that an order picker can access items from both sides of a walking aisle without needing to switch aisles.
  - **Ultra-narrow aisles:** In the shelfing area, aisles are ultra-narrow. Here, pickers must enter and leave an aisle from the same side, as the picker cart cannot traverse these aisles.

Customer orders

The incoming customer orders are the dynamic part of the problem and are defined by the set O, where each order $o \in O$ consists of a set of items, which corresponds to a subset of pick locations $P_o \subseteq L$ that must be visited.

The objective of the batch formation stage is to group the set of orders O into a set of batches B, where each batch $b \in B$ is a collection of one or more orders. For each batch, a corresponding set of unique pick locations $P_b \subseteq L$ is defined as the union of all pick locations from the orders within that batch.

For each batch b and its corresponding set of pick locations P_b, the picker routing problem must be solved. This problem focuses on finding a continuous tour that starts at the depot, visits all locations in P_b exactly once and ends at the depot, all while minimising the total travel distance.

To accurately model movement in the ultra-narrow aisles, an additional index t over each arc (i, j) is introduced, which represents a specific feasible travel path between nodes i and j. Let T_{ij} be the set of feasible travel paths from node i to j, and let $S = \{0,1\}$ be the set of entry sides of an ultra-narrow aisle, where 0 is the bottom and 1 is the top.

Each path $t \in T_{ij}$ has an associated travel distance $d_{i,j}^t$, and each node $i \in N_r$ has an assigned aisle entry side $s_i \in S$. Additionally, $a_i \in R$ represents the ultra-narrow aisle r to which node $i \in N_r$ belongs, allowing aisle-specific constraints to be applied.

A binary decision variable $x_{ij}^t \in \{0,1\}$ is used to indicate whether path $t \in T_{ij}$ is selected to travel from node $i \in N$ to node $j \in N$. Additionally, the binary decision variable $z_r^s \in \{0,1\}$ is introduced to enforce entry and exit consistency for each ultra-narrow aisle $r \in R$. Moreover, the auxiliary variable $u_i \in \mathbb{R}$ is used for subtour elimination.

Using this structure, the order picker process is represented as a graph traversal problem where the overall objective is to minimise the total travel distance required to visit all pick locations in P.

Problem taxonomy

Given the decomposition described above and the unique environmental constraint related to the mixed aisle structure, the problem solved in this thesis can be classified as a Two-Stage Order Picking Problem with Batching and mixed-layout Ultra-Narrow Aisles (TS-OPPB-UNA). Since the decisions for the batch formation and route optimisation problem are not taken jointly, it was chosen to not call the problem at hand a Joint Order Batching and Picker Routing Problem (JOBPRP), but rather a 'Two-Stage Order Picking Problem with Batching' (TS-OPPB). The UNA provides special attention to the unique constraints of the shelfing area in the warehouse layout.

4.2 Stage 1: batch formation

The first stage of the proposed solution approach is the batch formation. The main objective of this stage is to smartly group individual orders together into feasible batches. Since grouping orders with high spatial proximity can reduce the total travel distance and picking time significantly, this stage can have a large impact on the warehouse efficiency.

In this section, the logic and constraints governing the batch formation process are discussed. First, in Section 4.2.1, the key operational requirements and assumptions for forming a valid and feasible batch are provided. In Section 4.2.2, two algorithmic approaches for the batch formation are given, after which the structure of the output of this stage, and thus the input of the next stage, is provided and explained in Section 4.2.3.

4.2.1 Batching requirements and assumptions

For the first stage of the solution framework, a set of operational requirements and assumptions are established, to make the model as close as possible to the real-world situation, while still keeping it manageable and solvable.

Core batching requirements

These requirements define what forms a valid and feasible batch. Every batch that is formed must adhere to these requirements.

- **Order integrity**
  This is the most important requirement for the batch formation. A single customer order cannot be divided across different batches; all items belonging to a customer order must be picked within the same batch.
- **Picker cart capacity**
  While the physical capacity of the picker cart (in terms of weight or volume) is not explicitly modelled due to data limitations, it must be modelled due to the real-world complications. Therefore, the batch formation process is constrained by two different metrics: a maximum number of unique pick locations and a maximum total quantity of items per batch. By limiting these two values, the model prevents the creation of batches that would be physically impossible to handle.

Process requirements and assumptions

The following points relate to how formed batches are processed by an order picker and how these fit into the overall workflow of Company X.

- **Batch assignment**
  Each batch is assigned to a single order picker and must be completed by that individual without interruption or reassignment.
- **No overlapping batches**
  The model assumes that a batch must be fully completed before a new batch can be started by an order picker. This means that no overlapping or interleaving of batch execution is allowed.
- **Single tour per batch**
  The collection of all items within a single batch must be completed within one single tour. Each order picker starts at the depot, visits all required pick locations for that batch, and only then return to the depot. Intermediate returns at the depot are not allowed.
- **Route feasibility**
  All pick locations for a given batch must be reachable via a feasible path in the warehouse graph.

4.2.2 Batch formation algorithms

To fulfil the requirements outlined above, two different algorithms are implemented for the batch formation stage. Both algorithms aim to create feasible batches by grouping customer orders based on their spatial proximity, thereby aiming to reduce the potential travel distance required to pick all items within the batch. The first algorithm is a sequential greedy algorithm, while the second is a globally oriented Clarke & Wright savings algorithm. Below, both algorithms are discussed.

Greedy algorithm

The greedy batch formation algorithm forms batches sequentially by evaluating potential cost savings from combining orders. De Koster et al. (2007) describe this type of batching as *Proximity batching*, where "each order is assigned to a batch based on proximity of its storage location to those of other orders." For each batch, the first order is selected from the list of unassigned orders. Then, the algorithm evaluates the savings from adding an order to the current batch, where the savings are computed as follows:

$$Savings = Cost(Batch) + Cost(Order) - Cost(Batch \cup Order)$$

If the savings are positive, it is advantageous to add the order to the batch, since handling them separately would result in a higher total travelling distance. After all orders are evaluated, the order with the highest savings is added to the batch, after which it is taken away from the unassigned order list. This procedure continues until all orders are assigned to batches or the batch capacity is reached. In Algorithm 1, the pseudocode for the greedy algorithm is given.

Algorithm 1 Greedy Batch Formation Algorithm

Input: Order list $O = \{O_1, O_2, \ldots, O_m\}$, maximum batch size B_{max}, maximum number of pick locations L_{max}

Output: Set of batches $\mathcal{B} = \{B_1, B_2, \ldots, B_k\}$

```
B ← ∅
R ← O
while R ≠ ∅ do
    B ← new batch containing first order in R
    Remove selected order from R
    while True do
        bestSaving ← 0
        bestOrder ← None
        for all O ∈ R do
            Calculate total items q and unique locations l in B and O
            if q(B) + q(O) > Bmax or l(B) + l(O) > Lmax then
                continue
            end if
            CB ← ORDERCOST(B)
            CO ← ORDERCOST(O)
            Cmerged ← ORDERCOST(B ∪ O)
            savings ← CB + CO − Cmerged
            if savings > bestSaving then
                bestSaving ← savings
                bestOrder ← O
            end if
        end for
        if bestOrder = None then
            break
        else
            Add bestOrder to B
            Remove bestOrder from R
        end if
    end while
    Add B to B
end while
return B
```

Clarke & Wright savings algorithm

To address potential limitations of the greedy approach, the Clarke & Wright (CW) savings algorithm is adapted for the formation of batches. This algorithm is originally developed for the vehicle routing problem by Clarke and Wright (1964). The adapted savings algorithm is shown in Algorithm 2 and starts by forming separate batches for each order. For all feasible pairs of batches, the savings of merging them together are calculated, where the savings are calculated in the same way as in the greedy algorithm. After all potential savings are calculated, they are sorted in descending order. The pair with the largest savings is merged, but only if the merged batch respects the batch capacity constraints. This process of sequentially merging batches with the largest savings continues until no more merges with positive savings can be performed. The set of batches that is formed creates the final solution, where all batches respect the specific constraints.

Algorithm 2 Clarke & Wright Batch Formation Algorithm

Input: Order list $O = \{O_1, O_2, \ldots, O_m\}$, maximum batch size B_{max}, maximum number of pick locations L_{max}
Output: Set of batches $B = \{B_1, B_2, \ldots, B_k\}$
Initialize each order as its own batch: $B \leftarrow [O_1, O_2, \ldots, O_m]$
$S \leftarrow$ empty list
for all unordered pairs of batches (B_i, B_j), $i < j$ **do**
    $B_{ij} \leftarrow B_i \cup B_j$
    Calculate total items q_{ij} and unique locations l_{ij} in B_{ij}
    if $q_{ij} \leq B_{max}$ **and** $l_{ij} \leq L_{max}$ **then**
        $C_i \leftarrow$ ORDERCOST(B_i)
        $C_j \leftarrow$ ORDERCOST(B_j)
        $C_{ij} \leftarrow$ ORDERCOST(B_{ij})
        $s_{ij} \leftarrow C_i + C_j - C_{ij}$
        Add (s_{ij}, i, j) to S
    end if
end for
Sort S in descending order by s_{ij}
for all (s_{ij}, i, j) in S **do**
    if $s_{ij} > 0$ **and** B_i and B_j not yet merged **then**
        $B_{ij} \leftarrow B_i \cup B_j$
        Calculate q_{ij} and l_{ij} for B_{ij}
        if $q_{ij} \leq B_{max}$ **and** $l_{ij} \leq L_{max}$ **then**
            Merge B_i and B_j
        end if
    end if
end while
return B

In Chapter 5, both batch formation algorithms will be evaluated on their efficiency and the overall performance of the obtained batches.

4.2.3 Solution representation

The solution of the batch formation is represented by a list of batches $\mathbf{B} = \{B_1, B_2, \ldots, B_k\}$, where each batch B_i is a set of customer orders. Each order contains a set of pick locations in the warehouse, corresponding to the location of the item in the customer order.

Figure 4.2 shows the output structure of the batch formation process and an example solution representation. The top part of the figure shows a high level overview of the relationship between the input and output of the algorithm. The maximum batch size and all customer orders are processed to produce batches that can be used as input for the routing stage. The bottom part gives an example, where the input consists of five customer orders with a total of thirteen different pick locations and a maximum batch size of eight items. The output shows the batches both at the order level and as a list of item locations. It can be seen that B_1 contains orders 1, 2 and 5 to come to a total of six items in the batch. B_2 contains orders 3 and 4, which total at seven items. Here, the maximum size of a batch came into play, since orders 3 and 4 could not fit into B_1 anymore, so the only order that could be added was order 5. The list of pick locations is used in the route optimisation stage, whereas the list of orders per batch is used in the processing of the batch at the depot.

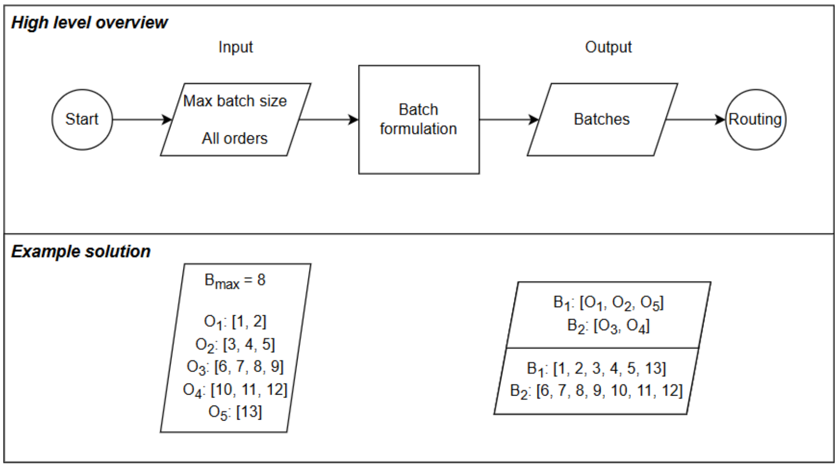


Figure 4.2 Solution representation of the batch formation

4.3 Stage 2: route optimisation

Once the batches have been formed in Stage 1, the second stage, route optimisation, starts. In this stage, the most efficient picking route for each individual batch is determined. An optimal or near-optimal tour must be calculated that starts at the depot, visits all pick locations in a batch, and then ends at the depot, all while respecting the physical and operational constraints of the warehouse environment.

This section presents two different approaches for solving the routing problem. First, a Mixed Integer Linear Programming (MILP) model is given in Section 4.3.1, which is detailed in a mathematical model. This MILP model provides a method for finding an optimal route and serves as a benchmark for validating the solution quality of the second approach. This approach, a metaheuristic approach based on Simulated Annealing (SA), is discussed in Section 4.3.2 and it is designed to be a quicker, more scalable and flexible alternative to the computational extensive MILP. The SA method is created to handle large and complex instances as they are encountered in the daily operations at Company X.

4.3.1 Mathematical formulation

In this section, a mathematical formulation is presented that represents the order picker problem at Company X. Here, the problem is modelled as a Mixed Integer Linear Program (MILP) aiming to determine an optimal picker route per predetermined batch. Given that the problem at hand is too complex and large to obtain a practical solution, the MILP model is used as a benchmark for the metaheuristic formulation that is presented in Section 4.3.2. Below, some MILP specific assumptions are described. Furthermore, the notations, mathematical model, and a toy example are detailed.

4.3.1.1 MILP specific assumption

Before presenting the mathematical model, an assumption specifically made for the MILP formulation must be addressed. This assumption relates to how the ultra-narrow aisles are handled in the MILP model:

- **Uniform ultra-narrow aisle entry**
  To reduce the combinatorial complexity of the MILP model, it is assumed that all ultra-narrow aisles in a picking route must be entered from the same side, so either all from the top or all from the bottom. This ensures that the model is at most solved twice for one batch, once for each uniform entry side, after which the best solutions is selected. It is important to note that this assumption is relaxed in the metaheuristic approach described in Section 4.3.2.

- **Single entry per aisle**
  It is also assumed that each ultra-narrow aisle is entered at most once per tour. This ensures that all pick locations within an ultra-narrow aisle are visited in sequence before the picker leaves the aisle.

4.3.1.2 Notation and definitions

This section defines the notation that is used in the MILP model. All relevant sets, parameters, and decision variables are included to formally describe the order picker routing problem.

SETS

L	Set of storage locations
W	Set of walking points
d	Depot node
N	Set of all nodes; $N = L \cup W \cup \{d\}$
A	Set of arcs; $A \subseteq N \times N$
P	Set of pick locations $P \subseteq L$
R	Set of ultra-narrow aisles
N_r	Set of nodes in ultra-narrow aisles; $N_r \subseteq N$
S	Set of aisle entry sides (0 = bottom, 1 = top)
T_{ij}	Set of feasible travel paths between nodes i and j
i, j	Indices to nodes $i, j \in N$
r	Indices to ultra-narrow aisles $r \in R$
t	Indices to feasible travel paths $t \in T$

PARAMETERS

d_{ij}^t	Distance of path $t \in T_{ij}$ between nodes $i \in N$ and $j \in N$
s_i	$\in S$: Entry side of node $i \in N_r$, only defined for ultra-narrow aisles
a_i	aisle r to which node $i \in N$ belongs, $r \in R$

DECISION VARIABLES

$$x_{ij}^t = \begin{cases} 1, & \text{if path t} \in T_{ij} \text{ from node } i \in N \text{ to node } j \in N \; is \; chosen \\ 0, & otherwise \end{cases}$$

u_i = Auxiliary variable for subtour elimination (Miller-Tucker-Zemlin formulation), used only if $i \in P$

$$z_r^s = \begin{cases} 1, & \text{if an ultra-narrow aisle } r \in R \text{ is entered from side } s \in S \\ 0, & otherwise \end{cases}$$

4.3.1.3 Mathematical model

This section presents the MILP model of the order picker routing problem. It includes the objective function and its constraints.

OBJECTIVE FUNCTION

$$\min \sum_{i \in N} \sum_{j \in N} \sum_{t \in T_{ij}} d_{i,j}^t * x_{ij}^t \tag{4.1}$$

CONSTRAINTS

Subject to:

$$\sum_{j \in N} \sum_{t \in T_{ij}} x_{dj}^t = 1 \tag{4.2}$$

$$\sum_{i \in N} \sum_{t \in T_{ij}} x_{id}^t = 1 \tag{4.3}$$

$$\sum_{j \in N} \sum_{t \in T_{ij}} x_{ij}^t = \sum_{j \in N} \sum_{t \in T_{ij}} x_{ji}^t \qquad \forall i \in N \ \{d\} \tag{4.4}$$

$$\sum_{j \in N} \sum_{t \in T_{ij}} x_{ji}^t = 1 \qquad \forall i \in P \tag{4.5}$$

$$u_i - u_j + |P| \cdot \sum_{t \in T_{ij}} x_{ij}^t \leq |P| - 1 \qquad \forall i, j \in P, i \neq j \tag{4.6}$$

$$u_d = 0 \tag{4.7}$$

$$\sum_{\substack{i \notin N_r \\ j \in N_r \\ s_j = s}} \sum_{t \in T_{ij}} x_{ij}^t \leq M \cdot z_r^s \qquad \forall r \in R \tag{4.8}$$

$$\sum_{\substack{i \in N_r \\ j \notin N_r \\ s_j = s-1}} \sum_{t \in T_{ij}} x_{ij}^t \leq M \cdot (1 - z_r^s) \qquad \forall r \in R \tag{4.9}$$

$$x_{ij}^t \in 0,1 \qquad \forall i, j \in N, t \in T_{ij} \tag{4.10}$$

$$u_i \in \mathbb{R} \qquad \forall i \in P \tag{4.11}$$

$$z_r^s \in 0,1 \qquad \forall r \in R, s \in S \tag{4.12}$$

The objective function (4.1) minimises the total travel distance of a picker route. Constraints (4.2) and (4.3) ensure that each picker route starts at the depot by guaranteeing that exactly one arc must leave the depot node and one arc must return to the depot node. Constraint (4.4) maintains flow continuity throughout the tour, ensuring that when a node is entered, it should also be left again. Constraint (4.5) ensures that each required pick location can and must only be entered once. The exiting of this node is ensured by the flow continuity constraint. Constraint (4.6) is the subtour elimination constraint, making sure that solutions consist of only one tour that connects all pick locations. Constraint (4.7) anchors the sequence of the tour at the depot by fixing the MTZ variable for the depot to zero. Constraints (4.8) and (4.9) ensure that if there is an entry into an ultra-narrow aisle from one side, the route must also exit the aisle from that side. Constraints (4.10) and (4.12) are integrality constraints that ensure that these decision variables are binary integers. Constraint (4.11) defines the MTZ variable as a continuous variable used for subtour elimination.

Together, these constraints ensure that all operational restrictions from Section 4.1 are satisfied; the predefined batch is completed in a single continuous tour, starting and ending at the depot while visiting all required pick locations.

4.3.1.4 Toy example

To illustrate the solution provided by the MILP model on a real-world configuration, a small routing instance is created based a predefined batch of products. The toy example shows that the output of the model is a valid and efficient picking route that satisfies all operational requirements.

The example is based on a batch that contains 15 different products, stored across different zones in the warehouse, including the deck zone, pallet racks, and the shelving area. Table 4.1 shows the products, their quantities, and the storage locations. The order of the products in the table is based on the alphabetical order of the locations. The picker must visit all these locations in one uninterrupted tour, starting and ending at the depot. The warehouse layout is represented by a graph, where the pick locations correspond to nodes, and the arcs represent the feasible walking paths.

The MILP model computes a route that visits all required pick locations and returns to the depot. In Figure 4.3 the result is shown graphically, where the green paths represent the route, the red markers indicate the required pick locations and the black square is the depot. Additionally, the picking index is shown for each pick location. The total distance of the route is 229 meter and it respects all modelling constraints. As can be seen, all ultra-narrow aisles are being entered from the bottom. The model also calculated the route with the entry side on the top, but since this would result in a longer route, it was decided that the entry side should be the bottom.

Table 4.1 Products to be picked in the batch

Product	Amount	Location
Product A	1	Deck.02.10.A
Product B	20	Pallet.01.01
Product C	1	Pallet.01.06
Product D	2	Pallet.01.08
Product E	2	Pallet.01.12
Product F	1	Pallet.01.26
Product G	1	Pallet.02.07
Product H	1	Pallet.04.29
Product I	1	Pallet.07.04
Product J	3	Pallet.07.13
Product K	1	Pallet.14.35
Product L	1	Shelf.12.7.3.B
Product M	2	Shelf.13.1.1.C
Product N	2	Shelf.13.3.2.A
Product O	1	Shelf.7.1.2.C

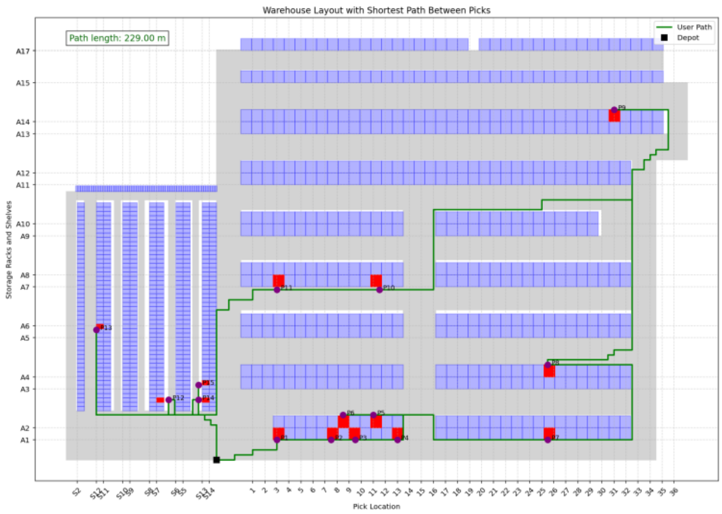


Figure 4.3 MILP solution for a batch of 15 unique items with total length of 229 meters

This toy example shows how the MILP model creates an optimal solution for a realistic batch in a relatively small warehouse instance. While the model successfully ensures a feasible and optimal solution, it must be noted that it becomes increasingly more time consuming and complex as the number of pick locations grow.

These limitations create the need for the development of the metaheuristic approach, which is capable of handling larger problem instances while also taking different entry sides for the ultra-narrow aisles into consideration.

4.3.2 Metaheuristic approach

While the MILP model provides an optimal benchmark, it is not suitable for the larger and more complex situation at Company X, due to it computational time. To provide a solution that is practical and scalable, a metaheuristic approach based on Simulated Annealing (SA) is developed. SA is chosen for its flexibility, its ability to escape local optima, and its proven effectiveness in exploring large solution spaces. This section discusses the specific routing requirements, the logic used for the SA algorithm and the way the outputted routes are evaluated

4.3.2.1 Routing requirements and assumptions

The main requirements of the routing process that were applicable to the MILP model are still relevant for the SA model. The main requirements are:

- Each route starts and ends at a depot
- All pick locations within a batch are reachable via feasible paths
- A picking route is a single tour, without interruptions or early returns to the depot

Although these requirements remain applicable, one of the MILP-specific assumptions that was introduced in Section 4.3.1 must be revised in relation to the SA model. For the MILP to work, it was assumed that an ultra-narrow aisle could only be entered once. Additionally, it was assumed that all ultra-narrow aisles should be entered from the same side, so either all aisles should be entered from the top or all from the bottom. These assumptions were required to keep the complexity of the model in check, while still allowing the model to handle the ultra-narrow aisles in a practical way.

In the SA approach, the assumption related to the uniform entry side is relaxed. Now, the model evaluates the entry side of an ultra-narrow aisle independently of the other aisles. This makes it possible for the route to enter one ultra-narrow aisle through the top side, while it enters another one via the bottom. However, it is still assumed that ultra-narrow aisles can only be entered once, which requires that all pick locations in such aisles are visited in sequence. This refinement provides additional flexibility in the routing process.

4.3.2.2 Routing logic

The goal of the route optimisation process is to obtain the picking sequence that results in a route with the minimal total travel distance required to visit all pick locations within a batch. To determine this picking sequence, a simulated annealing algorithm is employed, which is shown in Algorithm 3.

For each batch that is created in the batch formation process, an initial solution is generated. This is done by randomly arranging all the pick locations within the batch. Once this initial solution is created, the route is evaluated and the costs are saved. For each iteration of the algorithm, a candidate route is created by applying a neighbourhood operator, which are defined in Section 4.3.2.3. Although the sequence of pick locations is only modified in a small way, the effects on the total costs can be significant. If the candidate route is shorter than the current solution, it will be accepted immediately. If the new route is longer, it can still be accepted by the SA algorithm, but that depends on both the increase in distance and the current value of the temperature parameter. This mechanism allows the SA algorithm to escape local optima, since it occasionally accepts worse solutions, especially in the beginning of the run when the temperature parameter is still relatively high. As the run progresses, the temperature parameter gradually decreases according to a pre-set cooling schedule. The likelihood of accepting worse solutions becomes smaller with lower temperatures, thus shifting the focus from exploration of large solution spaces to exploitation of promising solutions. Once a final temperature is reached, the algorithm terminates. The best found solution is then selected as the final route for the batch.

Algorithm 3 Simulated Annealing for Route Optimisation

Input: Set of items in batch B, initial temperature T_0, stopping temperature T_{end}, cooling rate α, max iterations per temperature MaxIter

Output: Best solution found S_{best}, $Cost_{best}$

```
S ← GenerateInitialSolution(B)
Cost(S) ← EvaluateRouting(S)
S_best ← S
Cost_best ← Cost(S)
T ← T_0
while T > T_end do
    for i = 1 to MaxIter do
        S′ ← ApplyNeighbourhoodOperator(S)
        Cost(S′) ← EvaluateRouting(S′)
        Δ ← Cost(S′) − Cost(S)
        if Δ < 0 or Random(0,1) < exp(−Δ/T) then
            S ← S′
            Cost(S) ← Cost(S′)
            if Cost(S′) < Cost_best then
                S_best ← S′
                Cost_best ← Cost(S′)
            end if
        end if
    end for
    T ← α · T
end while
return S_best, Cost_best
```

The SA algorithm has some parameters that influence the effectiveness and the course of the algorithm. The values of these parameters should be tuned accordingly, which is done in Chapter 5.

4.3.2.3 Neighbourhood operators

Simulated annealing uses different operators to create new solutions based on the current solution. For the creation of different routes, three operators are used, which are commonly referred to as Swap, Insert and 2-opt.

- **Swap:** The swap operator selects two different pick locations at random and exchanges their positions in the route. These quick local rearrangements help in diversifying the solution space
- **Insert:** The insert operator takes a pick location from its position and inserts it at another position in the route. These changes can have more substantial impact on the route than swaps, thus improving the chance of escaping local optima.
- **2-opt:** The 2-opt operator takes a pair of non-contiguous paths and rearranges the connections between the four pick locations, effectively changing the order in which a segment of the picking route is traversed. This makes the operator effective at removing crossovers within a route, thus improving the solution.

During the route creation of the SA, two different strategies of applying the operators are used:

- **Random operator selection:** At each iteration of the SA algorithm, one of the three operators is chosen at random to generate a new solution. By adjusting the probabilities at which an operator is chosen, either diversification or intensification can be encouraged.
- **Structured operator selection:** Operators are applied in a structured order, by first applying the Swap operator, then the 2-opt operator and lastly the Insert operator. If the first operator

does not improve the current solution, the next operator is applied. If an improvement is found, or if all operators are tested, the newly improved solution is returned. The sequence then restarts at the first operator. This selection strategy accelerates convergence by systematically using local improvements. Additionally, it can make use of all operators in a sequence, efficiently exploring the neighbourhood for improvements.

In Chapter 5, both strategies will be evaluated on their overall performance and efficiency.

4.3.2.4 Route representation and evaluation

The solution of the route optimisation is represented by an ordered list of pick locations, beginning and ending at the depot. This ordered list shows the sequence in which the order picker should visit all the required pick locations in a batch. Besides the ordered list, a graph is created that shows the route graphically.

Each route is evaluated on the total travel distance for the order picker, which is computed as the sum of the paths between consecutive pick locations. Another metric that is used for the evaluation of a batch and corresponding route is the total picking time. This total picking time is computed as follows:

$$Total\ picking\ time = Walking\ time + Search\ time + Collection\ time$$

Where:

$$Walking\ time = \frac{Distance}{Walking\ speed}$$

$$Search\ time = Search\ time\ per\ pick\ location * Number\ of\ unique\ pick\ locations$$

$$Collection\ time = \ Collection\ time\ per\ item * Number\ of\ items\ to\ be\ picked$$

With this metric, not only the efficiency of the route can be assessed, but also the real-world impact on the order picking operations at Company X. Additionally, this metric will be used to perform different kind of experiments in the next chapter. By implementing different types of technologies, factors like search time per pick location and collection time per item are influenced. This way, different technologies can be evaluated on the increase in productivity and efficiency of the order picking process.

4.4 Conclusion

This chapter provides a comprehensive answer to the research question “How can a solution approach be designed to improve the efficiency of Company X’s order picking process?” First, a two-stage solution framework is created for the batching and routing problem at hand to address its high complexity. After the decomposition, the problem is formally defined and its key requirements and assumptions are established. Here, the problem is classified as a Two-Stage Order Picking Problem with Batching and mixed-layout Ultra-Narrow Aisles (TS-OPPB-UNA), providing special attention to the unique constraints of the mixed-layout warehouse.

The framework consists of two sequential stages:

1. Batch formation: Two algorithms are presented to group customer orders efficiently into batches, a Greedy algorithm and a Clarke & Wright Savings algorithm. Both of these algorithms create feasible batches that respect the order integrity and cart capacity constraints.
2. Route optimisation: For the routing problem, two approaches are proposed. First, a MILP model is formulated that provides exact, optimal solutions for small instances, which will

serve as a valuable benchmark. For larger instances, the model becomes impractical due to its long computational times. Therefore, a faster and flexible Simulated Annealing metaheuristic approach is designed. This approach aims to find high-quality solutions for larger batches, as encountered in the daily operations of Company X. Different neighbourhood operator strategies are included to ensure that the best picking routes can be determined.

In the next chapter, experiments are conducted to find the most effective batching algorithm and operator selection strategy. Additionally, the parameters of the SA algorithm are determined in the experimental phase to further improve the picking routes and thus the efficiency of the picking process. Once this is done, the performance of the solution approach is evaluated against the current operations and the MILP benchmark.

5. Experimentation and evaluation

This chapter gives answer to the fourth research question: *"How does the developed solution improve warehouse efficiency compared to the current situation?"*. For this, an experimental design is outlined in Section 5.1 after which the data instances are described in Section 5.2. The algorithms created in the previous chapter are then tuned by selecting the most effective batch formation and route optimisation strategies, which is done in Section 5.3. Here, the performance of the algorithms is benchmarked against the exact MILP model and the current practice at Company X. Lastly, in Section 5.4, a series of scenarios are analysed to explore the robustness of the solution approach under varying conditions.

5.1 Experimental design

The experimental framework used to evaluate the solution approaches developed in Chapter 4 is shown in Table 5.1. The experiments are designed to systematically assess the created batching and routing algorithms on their effectiveness, efficiency, and practical applicability for order picking at Company X. The framework of experiments is divided into two main groups: experiments for the setup of the algorithm and experiments to evaluate different scenarios.

<u>Algorithm setup</u>

The first group of experiments is focused on selecting and tuning the algorithmic components for batch formation and route optimisation. Here, experiments are conducted to evaluate which algorithm for the batch formation process is the best (Experiment 1) and which operator selection strategy should be chosen (Experiment 2). After this has been determined, the parameters of the simulated annealing algorithm are tuned, to ensure that the performance and convergence of the metaheuristic are optimal (Experiment 3). To validate the solution approach, the MILP model is compared to the SA model (Experiment 4). Here, the batches that are formed by the batch formation algorithm are used as predefined batches for the MILP. This way, the MILP and SA use the same batches for the route creation, which allows for a fair comparison between the two solution methods. The comparison is based on the travel distance of the created routes and the computation times it took both methods to create the routes. Additionally, the created routes are compared to how the orders have been processed in reality, highlighting the potential savings and improvements.

<u>Scenario evaluation</u>

The second group of experiments investigates how the optimised algorithm performs under different practical warehouse scenarios. The effects of the capacity constraints on total distance and picking times are investigated by experimenting with different batch sizes (Experiment 5). Experiment 6 evaluates the impact of technology: the introduction of scanners (Experiment 6.1) and automated guided vehicles (AGVs) (Experiment 6.2). The influence of the warehouse layout is assessed in Experiment 7, where the effects of removing the ultra-narrow aisles (Experiment 7.1) and changing the availability of the back cross aisle (Experiment 7.2) are examined.

This structure allows for the selection of the most optimal solution configuration, while also evaluating scenarios under different operational conditions.

Table 5.1 Experimental design

#	*Experiment*	*Description*	*Goal / purpose*
		Algorithm setup	
1	Batch optimisation method	Compare batching algorithms (Greedy vs. C&W) on distance, batch size and computation time	Select best batch formation algorithm
2	Operator selection strategy	Compare random vs. structured neighbourhood operator strategy	Select best selection strategy
3	SA Parameter Tuning	Tune the parameters of the simulated annealing algorithm using chosen batch algorithm and operator strategy	Improve performance and convergence speed of the metaheuristic
4	Algorithm performance analysis	Analyse the behaviour of the SA in relation to the MILP and with different sized problem instances	Benchmark SA against MILP and show scalability of model
		Scenario evaluation	
5	Maximum batch size	Compare total distance with different batch sizes during batch formation process	Examine which maximum batch size gives best trade-off between computation time and objective value
6.1	Picker support systems	Simulate the effect of picker support systems on the picking time	Assess potential efficiency gains from support systems
6.2	AGVs	Analyse the impact of implementing an AGV that transports items to depot	Evaluate potential efficiency gains of implementing AGVs
7.1	Widening of ultra-narrow aisles	Model the warehouse as if ultra-narrow aisles were removed by increasing shelf spacing	Measure the impact of layout redesign on routing efficiency
7.2	Close back aisle	Compare routes with the back aisle available or blocked	Measure how aisle layout flexibility impacts route length

To conduct the experiments and evaluate the results, a computer running Windows 11 and equipped with an Intel® Core™ i7-9750H processor (2.60GHz) and 16 GB of RAM is used. For the execution of both the MILP and SA model, Spyder 6.0.3 IDE integrated with Python 3.12.3 is used, where the optimisation tool Gurobi (version 12.0.1) is used to solve the MILP model.

5.2 Data instances

For the experiments, two types of datasets are used, synthetic and real-world. Three synthetic datasets are used for the setup of the algorithms (first three experiments in Table 5.1), while real-world data is retrieved from the WMS of Company X for the evaluation of the performance of the algorithm (Experiment 4) and the scenario evaluation. In Table 5.2, a summary of the key characteristics of each dataset is provided.

The three synthetic datasets (SD1, SD2 and SD3) represent extreme warehouse scenarios and are used for the setup of the algorithms. By ensuring that the algorithm works optimally in extreme cases, the risk of overfitting to operational data is minimised, while the robustness of the algorithm is improved. SD1 contains randomly selected pick locations which results in orders that are widely scattered throughout the warehouse. SD2 features orders that are concentrated within three pick aisles. With 80% of all orders in these three aisles, a dense picking environment is created, which allows for the testing of the algorithm's ability to exploit the proximity of orders. SD3 contains a high percentage (more than 50%) of pick locations within the ultra-narrow aisles. This directly stress-tests the specific routing logic within this area of the warehouse.

The real-world datasets (D1 – D5) are extracted from the WMS, each corresponding to a distinct working day (Monday to Friday). The datasets contain all the orders that are present at the beginning of the corresponding day to capture the peak order volume just before order picking operations are started. The datasets include information on the order composition, the locations of the included items, and the product quantities. The datasets vary in the overall complexity and size. For example, D1 contains all the orders that were present in the system on a Monday morning, representing a peak moment where orders have accumulated over the weekend. The remaining datasets capture regular daily volumes. Dataset D1 is included in the overall performance evaluation to assess the scalability of the solution approach, although it is excluded from the scenario analysis experiments, due to the long computation times required for optimising the large dataset. Therefore, datasets D2 through D5 are used for the analysis of different warehouse scenarios, since they represent more typical daily volumes at Company X.

Table 5.2 Key characteristics of the datasets

Dataset	*Number of orders*	*Unique pick locations*	*Total quantity of items*
SD1	130	174	272
SD2	100	94	225
SD3	113	139	262
D1	381	186	843
D2	134	107	255
D3	146	116	850
D4	115	98	229
D5	158	104	532

5.3 Algorithm setup

This section presents the setup and tuning of the proposed solution algorithm. The experiments that are designed to systematically evaluate and select the most effective methods for batch formation, route optimisation, and metaheuristic parameter settings.

Unless otherwise stated, each experiment is executed five times per dataset to mitigate the effects of randomness, after which the average results are presented and evaluated. For the experiments performed before the parameters are selected, a baseline configuration is applied, which is defined in Section 5.3.3. Key parameters for the batch formation algorithm are the maximum number of unique pick locations and the maximum number of items per batch. Based on an analysis of historical batches recorded by the WMS of Company X, these values are set to 25 and 100, respectively. In Section 5.4.1, further experiments will be conducted with these parameter values. For the calculation of the total picking time, additional parameters are required: the walking speed, search time per pick location, and the collection time per item. Unless otherwise stated, these values are set to the values as shown in Table 5.3. These values are based on practical observations and measurements at Company X. Since walking speeds and pick times vary between different employees and order profiles, the average values used are determined in consultation with the warehouse staff. For all metrics in the remainder of this study, distances are reported in meters and times in seconds.

Table 5.3 Baseline parameter values

Parameter	*Value*
Max number of unique pick locations per batch	25
Max quantity of items per batch	100
Walking speed	1.1 meter per second
Search time per pick location	5 seconds
Collection time per item	3 seconds

The goal of this section is to describe the process of tuning the algorithm, by providing a logical and numerical basis for each design choice. This ensures that the final algorithm is robust and practical, so that it can effectively improve the picking routes at Company X.

5.3.1 Batch formation algorithm

The first decision that needs to be made is what batch formation algorithm to apply, the greedy algorithm or the Clarke & Wright (C&W) savings algorithm. Both algorithms aim to form feasible batches by grouping customer orders together, but they differ in how these orders are selected and merged. To decide which algorithm to choose, different experiments are performed to analyse which algorithm has the best performance.

Since the batching formation only outputs formed batches without creating routes, there are no direct measures that can evaluate the best algorithm except the batching computation time. These computation times are shown in Table 5.4, where the lowest values per dataset are given in bold. It can be seen that the Clarke & Wright savings algorithm forms the batches quicker than the greedy algorithm. Where the greedy algorithm takes between around 475 and 750 seconds to form batches, the C&W savings algorithm takes around 310 to 675 seconds, depending on the dataset. In the case of SD3, the C&W algorithm formed the batches about 40% quicker than the greedy algorithm, indicating that it is more efficient in the process of batching orders together.

Table 5.4 Average batch formation times (s) for batching algorithms

Batch formation algorithm	*SD1*	*SD2*	*SD3*
Greedy algorithm	748.21	482.26	528.61
C&W savings algorithm	**673.01**	**415.19**	**309.57**

To evaluate the performance of the formed batches, the route optimisation algorithm is applied twice on the formed batches, using both the random and the structured operator selection strategy. The results of all four combinations are shown in Table 5.5, where the minimal values for each dataset are highlighted in bold. Here, the experiments are defined as follows:

- Exp. 1: Greedy batching + Random operators
- Exp. 2: Greedy batching + Structured operators
- Exp. 3: C&W batching + Random operators
- Exp. 4: C&W batching + Structured operators

When comparing the same type of operator selection strategy (Exp. 1 vs Exp. 3 and Exp. 2 vs Exp. 4), it can be noted that the batches formed by the C&W savings algorithm perform better than those formed by the greedy algorithm. For SD1 and SD3, the C&W batches consistently outperform the greedy batches in all metrics, although the number of batches are equal. Especially for SD3, the differences are big, with a difference of about 180 meters in the total travel distance and a difference of about 180 seconds in the computation times. This shows that for datasets containing extremely scattered orders or orders concentrated within the ultra-narrow aisles, the C&W savings algorithm performs better than the greedy batching algorithm. This can be explained by the ability of the C&W to exploit spatial savings across the entire set of orders, which makes it highly effective for these scattered and complex scenarios.

For SD2, the results are a bit different; even though the greedy algorithm forms one more batch than the C&W algorithm, the total distances and picking times are lower, although these differences are small. The computation times rely mostly on the type of operator strategy that is used, so no clear conclusions can be made based on the batching strategy for this dataset. The reason for the better performing greedy algorithm are likely due to the nature of the order set. Since most orders are densely clustered together in three picking aisles, the C&W's advantage of global optimisation is smaller, giving the greedy algorithm an advantage of batching orders together, although this advantage is small.

Table 5.5 Average results of batch formation algorithm and operator selection strategy

	SD1				*SD2*				*SD3*			
ID	*Dist. (m)*	*Pick Time (s)*	*Run Time (s)*	*# of batches*	*Dist. (m)*	*Pick Time (s)*	*Run Time (s)*	*# of batches*	*Dist. (m)*	*Pick Time (s)*	*Run Time (s)*	*# of batches*
Exp. 1	1741.6	3269.27	333.49	7	**890**	**1989.09**	**125.59**	5	1392.8	2767.18	604.72	6
Exp. 2	1775.6	3300.18	515.87	7	923.2	2019.27	203.75	5	1417.4	2789.55	888.29	6
Exp. 3	**1675.4**	**3209.09**	**327.79**	7	900.8	1993.91	127.98	4	**1214.2**	**2589.82**	**445.13**	6
Exp. 4	1709	3239.63	512.14	7	924.4	2015.37	186.74	4	1221.4	2596.36	704.92	6

Based on the results shown in both Table 5.4 and Table 5.5, it is concluded that the C&W savings algorithm is the best performing batching algorithm. This algorithm forms batches significantly quicker than the greedy batching algorithm and the batches perform generally better. Only in an extreme situation where most orders are clustered together, the greedy algorithm performs slightly better solutions, but this does not outweigh the significantly longer computation time to create these batches in the first place. Therefore, the C&W savings algorithm is selected as the batch formation method.

5.3.2 Operator selection strategy

Within the simulated annealing algorithm, two different strategies can be used for the selection of neighbourhood operators, namely a random and a structured operator selection strategy. To evaluate which of these performs the best, each of the strategies is combined with the batch formation algorithms. The results of the performed experiments are shown in Table 5.5. In the random operator strategy, the three operators all have an equal probability of being applied, which is equal to 1/3.

These experiments reveal that the random operator selection strategy provides better route solutions than the structured approach for both batch formation algorithms. Additionally, it is shown that the structured selection strategy takes considerably more time to optimise the routes than the random approach. This suggests that for the specific problem at Company X, the diversification of the random strategy is more effective than the intensification of the structured strategy.

As a result, the random approach is selected as the operator selection strategy. This approach can be tuned to promote either diversification or intensification by changing the probabilities at which different operators are being applied. To evaluate the optimal operator frequencies, additional experiments are conducted. In these experiments, the frequencies at which the three different operators are applied change. If an operator is favoured, it is applied 60% of the time while the other two operators are applied both 20% of the time. Additionally, there are three experiments where only a single operator is applied at a probability of 100%. The results of these experiments are shown in Table 5.6 and provide insight into the relative effectiveness of each operator combination. The smallest values per dataset are given in bold.

The results show that Exp. 4, which favours the insert operator, provides the lowest average distance and picking times for two datasets. However, when these values are compared to the results of Exp. 7, which only uses this specific operator, it can be seen that only using the insert operator performs way worse. This is similar to the swap operator; only using this operator is worse than using a combination of the three operators. However, only using the swap operator results in the shortest running times, although the achieved solutions also result in the largest travel distances and picking times.

There is little difference between favouring the 2-opt operator or purely using this operator, although the running time increases slightly when the operator is used 100% of the time. Exp. 1 shows the most balance between the travel distance and the running time. It is always close to the shortest found travel distance, while achieving this in relative short running times. Because of this, a probability of 1/3 for each neighbourhood operator is selected as the most appropriate configuration for the random operator strategy.

Table 5.6 Average results of operator frequency experiments

	SD1			*SD2*			*SD3*		
Experiment	*Dist. (m)*	*Pick Time (s)*	*Run Time (s)*	*Dist. (m)*	*Pick Time (s)*	*Run Time (s)*	*Dist. (m)*	*Pick Time (s)*	*Run Time (s)*
Exp. 1: All equal	1675.4	3209.09	327.79	**900.8**	**1993.91**	127.98	1214.2	2589.82	**445.13**
Exp. 2: Favour 2-opt	1675.6	3209.27	336.20	901.0	1994.09	126.67	1214.8	2590.36	448.68
Exp. 3: Favour swap	1675.8	3209.46	330.12	903.2	1996.09	128.83	1215.2	2590.73	460.32
Exp. 4: Favour insert	**1675.2**	**3208.91**	333.91	903.2	1996.09	130.81	**1214.0**	**2589.64**	449.51
Exp. 5: Only 2-opt	1676.6	3210.18	342.10	901.0	1994.09	132.58	**1214.0**	**2589.64**	484.94
Exp. 6: Only swap	1726.4	3255.46	**326.09**	958.6	2046.46	**115.58**	1246.6	2619.27	450.25
Exp. 7: Only insert	1694.6	3226.55	331.66	915.0	2006.82	121.97	1224.2	2598.91	461.85

5.3.3 SA parameter selection

The final step in the setup of the algorithm is to tune the parameters of the simulated annealing algorithm. To ensure a high performing SA, experiments are conducted to test different configurations of initial and final temperature, cooling rate, and number of iterations per temperature step. The results of these experiments are summarised in Table 5.7, in which the lowest value per metrics is highlighted in bold.

Exp. 1 is the baseline configuration, which has the following values for the SA parameters:

- Initial temperature: 500
- Final temperature: 0.5
- Cooling rate: 0.025
- Iterations: 400

This configuration is selected based on initial parameter testing during the creation of the SA algorithm. This baseline configuration is also used for all experiments in Sections 5.3.1 and 5.3.2. For the experiments in Section 5.3.3, it serves mostly as a reference point for evaluating improvements.

For most of the other experiments, the parameters can take the following values, which are based around the values of the baseline configuration:

- Initial temperature: [250, 500, 750]
- Final temperature: [5, 0.5, 0.05]
- Cooling rate: [0.05, 0.025, 0.01]
- Iterations: [200, 400, 800]

In Exp. 2 to 9, only one parameter changes in comparison to the baseline configuration to see what the impact of this particular parameter is. In the other experiments, combinations of parameters are changed, to evaluate interactions between parameters. In the experiments names, it is stated which parameters are changed. In the last four experiments, extreme values for the parameters are taken, to see how the SA algorithm performs with parameters outside the predefined range.

The results indicate that Exp.12 and 19 achieve the best solutions in terms of distance and picking time across all datasets. Both experiments achieve the minimal results in two of the three datasets, although this comes at the cost of high running times. Many other experiments achieve very similar results, with the average distance varying slightly (less than a meter) from the minimal values found. From the configurations that achieve these near-optimal results, Exp. 16 achieves these results the quickest, on average. This shows that the configuration of this experiment provides a good trade-off between solution quality and computation time. The results are consistent over the three datasets, showing that the configuration is robust.

Therefore, the following values are selected as the final configuration for the SA parameters:

- Initial temperature: 750
- Final temperature: 0.05
- Cooling rate: 0.025
- Iterations: 400

Table 5.7 Average results of parameter tuning experiments

	SD1			*SD2*			*SD3*		
Experiment	*Dist. (m)*	*Pick Time (s)*	*Run Time (s)*	*Dist. (m)*	*Pick Time (s)*	*Run Time (s)*	*Dist. (m)*	*Pick Time (s)*	*Run Time (s)*
Exp. 1: Baseline (Best H+O)	1675.4	3209.09	327.79	900.8	1993.91	127.98	1214.2	2589.82	445.13
Exp. 2: Initial temp - 250	1675.0	3208.73	329.43	903.2	1996.09	130.72	1214.8	2590.37	511.39
Exp. 3: Initial temp - 750	1675.0	3208.73	334.00	901.0	1994.09	135.72	1214.8	2590.37	465.94
Exp. 4: Final temp - 5	1842.4	3360.91	310.40	1124.4	2197.18	120.25	1345.6	2709.27	409.23
Exp. 5: Final temp - 0.05	1674.6	3208.37	362.32	**900.0**	**1993.18**	135.75	1212.2	2588.00	490.04
Exp. 6: Cooling rate - 0.05	1678.6	3212.00	300.63	905.2	1997.91	115.02	1215.8	2591.27	414.03
Exp. 7: Cooling rate - 0.01	1674.2	3208.00	456.77	900.2	1993.36	174.90	1213.0	2588.73	612.35
Exp. 8: Iterations - 200	1677.2	3210.73	305.10	905.2	1997.91	114.63	1216.0	2591.46	388.59
Exp. 9: Iterations - 800	1675.0	3208.73	412.28	900.4	1993.54	147.72	1213.8	2589.46	535.44
Exp. 10: Initial 250 & Iter. 200	1681.4	3214.55	**291.86**	905.6	1998.27	119.81	1215.8	2591.27	422.99
Exp. 11: Initial 750 & Iter. 800	1675.8	3209.46	421.56	901.0	1994.09	140.04	1213.8	2589.46	519.44
Exp. 12: Final 0.05 & Cool. 0.01	**1674.0**	**3207.82**	507.75	900.2	1993.36	178.09	1212.0	2587.82	650.01
Exp. 13: Initial 750 & Cool. 0.01	1674.8	3208.55	468.68	900.4	1993.54	163.81	1213.4	2589.09	577.86
Exp. 14: Initial 750 & Iter. 200	1677.2	3210.73	309.19	912.0	2004.09	**106.35**	1216.6	2592.00	**383.85**
Exp. 15: Initial 250 & Iter. 800	1674.4	3208.18	396.09	900.8	1993.91	150.08	1213.8	2589.46	505.10
Exp. 16: Initial 750 & Final 0.05	**1674.0**	**3207.82**	373.72	901.0	1994.09	133.06	1212.4	2588.18	467.28
Exp. 17: Initial temp - 1000	1676.6	3210.18	355.10	902.0	1995	126.49	1215.6	2591.09	439.85
Exp. 18: Final temp - 0.01	1675.2	3208.91	391.45	900.8	1993.91	139.80	**1212.0**	**2587.82**	512.90
Exp. 19: Cooling rate - 0.005	**1674.0**	**3207.82**	629.36	**900.0**	**1993.18**	210.47	1213.0	2588.73	849.54
Exp. 20: Iterations - 1200	**1674.0**	**3207.82**	545.48	900.2	1993.36	195.36	1213.0	2588.73	657.86

5.3.4 Algorithm performance analysis

Before the performance of the solution approach can be evaluated, a brief overview of the algorithm is provided. For the batch formation method, the Clarke & Wright (C&W) savings algorithm is selected for its ability to form high quality batches in an efficient way. The random operator strategy with equal probabilities of 1/3 is chosen for the neighbourhood selection method, given its good solution quality and computational efficiency. Additionally, it provides a balance between diversification and intensification of the solution space during the simulated annealing algorithm. The final simulated annealing parameter configuration that is selected is as follows: An initial temperature of 750, final temperature of 0.05, cooling rate of 0.025, and 400 iterations per temperature step. This combination provided robustness and high performance. In Table 5.8, an overview of the algorithm setup is given.

Table 5.8 Overview of algorithm setup

Description	*Final result*
Batch formation	Clarke & Wright savings algorithm
Operator selection	Random operator strategy
Operator frequency	[1/3, 1/3, 1/3]
Initial temperature	750
Final temperature	0.05
Cooling rate	0.025
Iterations	400

5.3.4.1 Performance evaluation

Now that the algorithm is tuned, its performance can be evaluated. For this, the SA algorithm is benchmarked against the MILP model, which can provide optimal routes for each batch. To evaluate the performance of both the SA algorithm and the MILP model, the algorithms are tested on the five real-world datasets (D1-D5). Since the MILP model does not perform batch formation, it requires predefined batches as input. To make a controlled comparison, the batches formed by the C&W savings algorithm of the solution approach are used as input to both the models. In Table 5.9, an overview of the created batches per dataset are shown.

Since both the SA and MILP solve identical picking problems, this setup provides a strong basis for comparison. However, this setup only evaluates the route optimisation capabilities of the SA, not the batch formation performance of the algorithm, but this performance is already evaluated in Section 5.3.1.

Table 5.9 Overview of batch formation per dataset

Dataset	*Avg batch creation time (s)*	*# of batches*
D1	3475.52	13
D2	522.79	5
D3	534.20	6
D4	389.48	5
D5	600.79	8

Before the results of the different models are analysed, it needs to be repeated that both models handle the ultra-narrow aisles in slightly different ways. In the MILP, all ultra-narrow aisles in a route are forced to be entered from the same side. The model computes the optimal route twice, once with the entry side at the top and once at the bottom. After computing both routes, it returns the shortest one found. The SA model, on the other hand, selects the entry side for each ultra-narrow aisle separately. This allows for a more flexibility, enabling the metaheuristic to fine more efficient solutions in specific cases. However, since every entry of an ultra-narrow aisle must be considered independently, the route optimisation must consider 2^n potential combinations of entry sides, where n is the number of ultra-narrow aisles in a given route.

Table 5.10 summarises the results of the comparison between the SA and the MILP for the different datasets. The MILP is tested under both short (60s) and long (1800s) time limits per entry side, to find both the operational and theoretical performance of the model. A time limit of 60 seconds means that the model runs 60 seconds to find the optimal solution with the entry side at the top of the ultra-narrow aisles and then 60 seconds with the entry side at the bottom, resulting in a maximum total running time of 120 seconds. For the long time limit, this totals to a maximum run time of 3600 seconds. In Appendix B, the results for each batch per dataset are shown, including the distance, optimality gap and running time per batch.

The results show that when the MILP model finds a solution, it always finds the optimal solution, since the optimality gap is 0.0%. However, sometimes the MILP cannot find feasible solutions within the given time limit, leaving the batch without a route. In Table 5.10, this can be seen for D1 and D5, depicted by a *. Here, the MILP model could not find feasible solutions in the given time limit, so the total travel distance does not show the full distance over all batches, but merely the total distance over all feasible routes.

When looking at the individual routes per batch, it can be seen that the MILP and SA almost always get the same results. However, sometimes the SA provides shorter routes than the MILP, even though the found solution is optimal. This is due to the ability of the SA model to navigate through the ultra-narrow aisles more flexible. By allowing the route to enter and exit each ultra-narrow aisle individually, more efficient batches can be created. This results in some differences between the results of the SA and the MILP.

It can also be seen that the MILP model solves some batches relatively quickly, especially simple routes that do not contain any picks within ultra-narrow aisles. The MILP model can solve these problems in a couple of seconds, whereas the SA model takes up a bit more time. However, for more complex instances, the SA model retrieves solutions more quickly. Additionally, the SA model consistently completes all batch optimisations, whereas the MILP model sometimes cannot find a feasible solution, even within the long time limit. The total running times of the SA model is a bit higher than the total running time of the MILP under a short time limit, although the MILP under a long time limit takes significantly more time to find solutions for all batches.

In summary, while the MILP model often provides optimal solutions, it cannot always find feasible solutions. The SA model always provides high quality solutions within a reasonable time, especially when compared to the MILP model under a long time limit. This shows that the SA algorithm is highly suitable for practical and complex operations at Company X.

*Table 5.10 Total distance and run time for all batches per dataset for Simulated Annealing (SA) and MILP (at 60s and 1800s time limits). A * indicates that the MILP did not find feasible solutions for all batches, so the shown distance is a sum over batches with solutions only.*

Method	*Metric*	*D1*	*D2*	*D3*	*D4*	*D5*
SA	Distance (m)	1947	946	857	1107	1209
	Run time (s)	796.60	357.76	524.71	486.96	318.07
MILP (60 s)	Distance (m)	1018*	1002	856	1291	1071*
	Run time (s)	805.74	207.09	209.91	225.37	241.20
MILP (1800 s)	Distance (m)	1920*	1002	856	1237	1204
	Run time (s)	11402.38	1939.88	1946.80	2398.45	2251.05

5.3.4.2 Benchmark against real-world

For further analysis of the performance of the solution approach, the results are compared to the routes as performed at Company X. Since the datasets are coming out of the WMS, the original batches can be evaluated on their routes.

In Table 5.11, the travel distances of the real-world routes are shown across the five datasets. These datasets represent the total walking distance walked by the warehouse staff when they walk the batch in the order that is presented by the WMS. These values can be compared to the results in Table 5.10 to evaluate the improvement of the solution approach to the real-world situation at Company X.

For all datasets, the developed solution approach creates picker routes with significantly shorter distances than the original batches. For example, in D1, the SA approach resulted in a total travel distance of 1947 meters, whereas the real-world distance totals at 4471 meters, so the SA reduces the distance with more than 50%. For the other datasets, similar reductions are observed.

This significant reduction can be explained by the large reduction of the number of batches that are required to process all the customer orders. In practice, there are a lot of batches created to retrieve all the different orders. Some of these batches only contain one customer order, resulting in highly inefficient routes. Additionally, there are batches that contain the same products, but since they are not grouped together in the same batch, the pick location has to be visited multiple times. By grouping the orders together based on their proximity, the batches formed by the formation algorithm do not have this problem. Together with the optimisation of the routes, significant improvements can be achieved from the current practice.

Table 5.11 Distances (m) of the real-world routes for all original batches

Dataset	*B1*	*B2*	*B3*	*B4*	*B5*	*B6*	*B7*	*B8*	*B9*	*B10*	*B11*	*B12*	*B13*	*B14*	*B15*	*B16*	*B17*	*B18*	*B19*	*Total*
D1	384	351	502	296	225	249	190	44	152	265	211	190	281	139	110	135	351	161	235	4471
D2	403	272	238	126	142	256	195	162	252											2046
D3	263	197	109	147	43	139	174	162	54	95	244	295								1922
D4	421	165	119	159	187	69	500	267	370	98										2355
D5	176	360	169	111	136	246	334	198	226	106	67	331	345							2805

5.4 Scenario evaluation

With the batching and routing algorithms optimised and evaluated under baseline operational conditions, it is essential to test how the solution approach performs under varying practical scenarios. Warehouse environments are constantly subject to changes, ranging from different operational constraints to the adoption of new technologies or new layout plans. To ensure that the developed algorithms are robust and provide practical value, this section entails a systematic evaluation across three scenario groups.

First, the influence of the batch size on the performance of the algorithms is evaluated. Then, the impact of implementing new technologies is assessed and discussed. Finally, different warehouse scenarios are evaluated, where current constraints related to the warehouse layout are either relaxed or tightened.

For the evaluation of these different scenarios, datasets D2 to D5 are used. Dataset D1 is left out of this analysis, due to the large computation times for the batch formation.

5.4.1 Batch size

One aspect that affects the operational efficiency of the order picking process is the size of batches. The values of the parameters defining the batch size have been based on analysis of historical batches. To investigate the effects of batch size on operational efficiency and route quality, a series of experiments are conducted that systematically evaluate the impact that both the maximum number of unique pick locations and the maximum quantity of items have. Table 5.12 summarises the main results of each experiment across the two datasets. Here, Exp. 0 represents the baseline with a batch size of 25 locations and 100 items.

The results show that different sized batches directly impact the number of batches required to fulfil all orders. However, the batching time shows that, even though the number of batches differs for the experiments, the computation time stays relatively constant, indicating that the batch formation process is not dependent on the number of total batches, but only on the number of individual customer orders present.

It is shown that across all datasets, reducing the batch size, either by lowering the number of locations (Exp. 1) or by lowering the maximum number of items (Exp. 3), leads to a substantial increase in the number of batches. This results in an increase in the total travel distance and picking time, since orders are grouped less efficiently and routes become more fragmented. For example, in D5, lowering the maximum locations from 25 to 15 almost doubles the number of batches required to pick all orders, which increase the travel distance by over 450 meters and the picking time by almost 600 seconds. Although smaller batches result in longer total travel distances and picking times, it should be noted that these batches can be optimised quicker, sometimes even by more than half.

By allowing more pick locations within a single batch (Exp. 2), the number of batches generally reduces, which in turn results in a lower travel distance, as the batches are more consolidated. Less unproductive travel is needed when a batch contains more different pick locations, although the optimisation becomes much more complicated, as can be seen in the running time. This shows that there is a trade-off between the travel distance and running time. Increasing the maximum number of items per batch (Exp. 4) to 150 has less significant impact, since the number of batches remains mostly unchanged compared to the baseline. This suggests that the limiting factor in these datasets is mostly the number of locations rather than the number of items in a batch.

In summary, the results show that batch size tuning has a direct effect on both the route quality and computational effort. Larger batches can sometimes improve travel distance and reduce the number of required routes but are harder to optimise, while other times they do not impact the travel distance significantly and can be optimised quicker. Therefore, careful tuning of batch parameters is recommended. Ideally, the size of a batch should be determined on the operational capacity constraints of the picking cart and volume of items. These are currently unknown at Company X but should be addressed in future works.

Table 5.12 Effects of varying batch sizes on batching and routing performance

Dataset	*Experiment*	*Locations / Items*	*Avg batching time*	*Number of batches*	*Avg distance*	*Avg picking time*	*Avg running time*
D2	Exp. 0	25 / 100	511.37	5	960.8	2233.46	364.86
	Exp. 1	15 / 100	517.64	9	1341.0	2639.09	243.17
	Exp. 2	35 / 100	518.47	4	854.2	2116.54	539.32
	Exp. 3	25 / 50	514.07	6	1012.0	2300.00	333.88
	Exp. 4	25 / 150	514.36	5	963.0	2235.46	367.41
D3	Exp. 0	25 / 100	534.20	6	859.2	4011.09	531.46
	Exp. 1	15 / 100	543.21	9	1079.8	4221.64	195.03
	Exp. 2	35 / 100	549.16	6	868.2	4034.27	401.96
	Exp. 3	25 / 50	500.02	10	1141.8	4353.00	251.76
	Exp. 4	25 / 150	541.56	6	835.4	3949.45	433.66
D4	Exp. 0	25 / 100	389.48	5	1117.6	2258.00	480.83
	Exp. 1	15 / 100	379.20	8	1295.2	2429.45	169.94
	Exp. 2	35 / 100	343.60	3	964.6	2083.91	712.47
	Exp. 3	25 / 50	356.38	5	1135.0	2288.82	595.31
	Exp. 4	25 / 150	366.09	5	1111.2	2252.18	499.69
D5	Exp. 0	25 / 100	600.79	8	1239.6	3662.91	290.57
	Exp. 1	15 / 100	633.07	15	1694.8	4246.73	127.63
	Exp. 2	35 / 100	644.08	6	1038.4	3430.00	569.53
	Exp. 3	25 / 50	585.63	11	1383.2	3943.46	275.49
	Exp. 4	25 / 150	613.08	7	1081.8	3434.45	318.90

5.4.2 Technological implementation

Innovative technologies can potentially improve the efficiency and effectiveness of the order picking process in warehouses. In this section, two different categories of technologies are considered and evaluated. First, the impact of different picking support systems is evaluated based on the improvement of the total picking time. Then, the use of automated guided vehicles (AGVs) is evaluated, which have the potential to further streamline the picking process. The analysis of these technologies is based on scenario experiments to isolate the effects on operational performance.

5.4.2.1 Picking support systems

Within a warehouse, the choice of picking technology has a direct and significant effect on the order picking efficiency. In this section, two relevant technological scenarios are evaluated: the implementation of barcode scanners and the adaption of pick-by-light systems. For each both scenarios, the picking time parameters (search time per pick location and collection time per item) are adjusted, reflecting the improved performance of these technologies. Barcode scanners result in a reduction of approximately 20% in search time and a 10% reduction in collection time when compared to the standard method. For the pick-by-light method, these reductions increase to roughly 70% and 40% for the search time and collection time, respectively. Empirical findings of Battini et al. (2015) show values consistent to those mentioned. They even report that the search time for the pick-to-light system becomes close to 0, as the lights immediately show which products to pick. For this study, it was decided to not set the search time to 0, since it becomes impractical to apply the pick-to-light system for all pick locations in the warehouse, so the chosen value is set as an average search time between the pick-to-light pick locations and the normal pick locations.

For the scenario analysis, no new batches are formed or optimised, since the implementation of new technologies does not impact the routing of the batches. The only metric that changes is total picking time required to finish all batches. The results of implementing new picking technologies are shown in Table 5.13.

Table 5.13 Impact of picking technology on total picking time

	Parameter values		*D2*	*D3*	*D4*	*D5*
Technology	*Search time per location*	*Collection time per item*	*Avg picking time*	*Avg picking time*	*Avg picking time*	*Avg picking time*
Standard (paper based)	5	3	2233.45	4011.09	2255.00	3662.91
Barcode scanner	4	2.7	2037.95	3620.09	2075.60	3315.31
Pick-by-light	1.5	1.8	1510.95	2515.09	1592.90	2366.51

The results show that the use of barcode scanners reduces the average picking time with about 10% across all datasets. The pick-by-light technology yields a much bigger reduction, with the total picking time being reduced by about 35% compared to the standard picking process. These findings are consistent with literature and show that investments in advanced picking support system can have substantial benefits, especially when efficient batch formation and route optimisation algorithms are applied besides them.

Other innovative picking systems, such as pick-to-voice and augmented reality technologies, can also be used to reduce search and collection times, as discussed in Section 3.4. While this study focuses on barcode scanners and pick-by-light systems due to their large practical relevance for Company X, the same scenario analysis can be applied to investigate the effects of these other technologies. For this, the picking time parameters need to be further adjusted.

5.4.2.2 Automated Guided Vehicles

In warehouses, Automated Guided Vehicles (AGVs) can be used for the transportation of goods from the picking area to the depot, where the items can be processed further. By integrating AGVs, the order picker can stay in the picking area, without needing them to make the unproductive way back to the depot. There are three experiments considered, Exp. 0 showcases the baseline, where no AGVs are implemented, so the order picker has to return back to the depot after each batch. Exp. 1 introduces the use of AGVs, but the batches are created under the normal conditions where no AGVs are present. Lastly, in Exp. 2, new batches are created based on the new situation. In Table 5.14, the results of the experiments are shown.

The results show a direct reduction of the travel distance when AGVs are introduced (Exp. 1). This reduction is a direct result of removing the path required to go back to the depot. Over all datasets, reduction of the travel distance between 15% and 20% are achieved. Because of this, the picking time is also reduced consequently. However, the running time remains mostly the unchanged, since the problem size does not change.

When new batches are formed with the AGV operations being considered (Exp. 2), further improvements are yielded in most datasets. The largest reduction is found in D5, where the travel distance decreases with 29%. The introduction of the AGVs also result in less complex problems, as both the batching time and route optimisation time reduce significantly for all datasets.

When looking at D4, it can be noted that forming new batches based on the AGV situation increases number of batches from 5 to 7, which results in a higher travel distance compared to Exp. 1, although it remains lower than the baseline level. This increase is due to the splitting of different batches into multiple smaller batches to maintain route compactness, although this results in a slightly larger travel distance.

Table 5.14 Results of implementing the use of an AGV

Dataset	*Experiment*	*Avg batching time*	*Number of batches*	*Avg distance*	*Avg picking time*	*Avg running time*
D2	Exp. 0: Baseline	522.79	5	960.8	2233.46	364.86
	Exp. 1: Original batch	522.79	5	833.8	2118.00	372.93
	Exp. 2: New batch	367.53	5	798.8	2071.18	368.88
D3	Exp. 0: Baseline	534.20	6	859.2	4011.09	531.46
	Exp. 1: Original batch	534.20	6	720.1	3884.64	542.21
	Exp. 2: New batch	377.87	6	681.2	3789.27	367.69
D4	Exp. 0: Baseline	389.48	5	1117.6	2258.00	480.83
	Exp. 1: Original batch	389.48	5	926.8	2084.54	456.20
	Exp. 2: New batch	247.31	7	1040.5	2182.91	365.46
D5	Exp. 0: Baseline	600.79	8	1239.6	3662.91	290.57
	Exp. 1: Original batch	600.79	8	991.6	3437.45	309.66
	Exp. 2: New batch	403.99	8	880.2	3246.18	208.28

Overall, the results show the operational advantage of AGVs in reducing both the travel distance and picking time. While introducing AGVs result in large potential gains, the magnitude of the improvement depends on the dataset's order structure, with the largest improvements seen in scenarios where return legs form a substantial portion of the total travel distance.

5.4.3 Warehouse scenarios

This section investigates what the effects of changing the warehouse layout of Company X are on the performance of the batching and routing algorithm. Here, two scenarios are evaluated, one where the ultra-narrow aisles are removed by widening them, so that order pickers can take the picking cart with them into the aisles. This allows them to fully traverse the aisles without the need to leave the aisles at the same side they entered. In the second scenario, the back cross aisle is closed, to investigate what impact it has if this cross aisle is fully closed off by using it as storage space. For both scenarios, experiments are conducted with both the original batches, as they are created for the baseline layout, and new batches specifically formed for the new warehouse layout.

5.4.3.1 Widening of ultra-narrow aisles

In this scenario, the ultra-narrow aisles in the shelving area are removed from the structural layout of the warehouse. This simulates the situation in which the aisles are widened in the real-world, so that the aisles are wide enough to allow an order picker to take their picking cart within the aisle. This removes the constraint that order pickers need to exit the ultra-narrow aisle on the same side they entered it, simplifying movement within the warehouse. To evaluate the impact that the ultra-narrow aisles have on the performance of the algorithms, three experiments are conducted. Exp. 0 acts as the baseline, where the ultra-narrow aisles are still in operation. In Exp. 1, the ultra-narrow aisles are removed, but the original batches, designed for a warehouse with the constraint, are now evaluated with the new warehouse structure. Lastly, in Exp. 2, new batches are created based on the altered layout. These three experiments provide a clear overview of how ultra-narrow aisle impact the batches and routes within a warehouse. The results of the experiments are shown in Table 5.15.

When the ultra-narrow aisles are removed in Exp. 1, the distance and picking time reduce for all datasets, even though the batches were created for a warehouse structure with the ultra-narrow aisles. The run time also drops drastically, which shows that the optimising of the routes is simplified. Without the ultra-narrow aisle constraint, the algorithm does not have to compute the routes for all possible entry sides of each specific aisle, significantly reducing the complexity of the routing process.

New batches based on a warehouse structure without ultra-narrow aisles (Exp. 2) have somewhat different results. In D2 and D3, the number of batches increase by 1. In D2, this results in an increase in both the distance and the picking time when compared to the baseline situation where there are ultra-narrow aisles. However, in D3, the extra batch does not result in an increase of the distance and picking time, suggesting that the new batches are better grouped together in the new warehouse structure. For D4 and D5, the newly formed batches reduce the total distance and picking time when compared to the baseline situation, although these reductions do not differ too much from the reductions that occur when analysing the original batches. In all datasets, the average running times are much shorter than the baseline experiment, emphasising that the routing process becomes simpler when the ultra-narrow aisles are widened. The creation of one additional batch results in more distributed batches that are not reaching the maximum batch size, which lowers the picking efficiency. In both datasets, the average running times are much shorter than the baseline experiment, emphasising that the routing process becomes simpler when the ultra-narrow aisles are widened.

Table 5.15 Results of widening the ultra-narrow aisles

Dataset	*Experiment*	*Avg batching time*	*Number of batches*	*Avg distance*	*Avg picking time*	*Avg running time*
D2	Exp. 0: Baseline	522.79	5	960.8	2233.46	364.86
	Exp. 1: Original batch	522.79	5	927.8	2203.45	51.75
	Exp. 2: New batch	534.31	6	1009.2	2317.45	47.53
D3	Exp. 0: Baseline	534.20	6	859.2	4011.09	531.46
	Exp. 1: Original batch	534.20	6	817.6	3973.27	51.79
	Exp. 2: New batch	539.95	7	818.8	3979.36	52.24
D4	Exp. 0: Baseline	389.48	5	1117.6	2258.00	480.83
	Exp. 1: Original batch	389.48	5	998.8	2150.00	44.29
	Exp. 2: New batch	348.70	5	999.2	2150.36	42.76
D5	Exp. 0: Baseline	600.79	8	1239.6	3662.91	290.57
	Exp. 1: Original batch	600.79	8	1182.2	3610.73	59.90
	Exp. 2: New batch	619.06	8	1201.2	3628.00	61.09

When batches are formed based on the new warehouse structure, the effects of the removal of the ultra-narrow aisle become case-specific, depending strongly on the underlying order structure and how the batch algorithm can exploit the new flexible warehouse structure. Overall, the running times decrease significantly when the aisles are widened, since the algorithm does not have to optimise many different combinations of entry side openings.

5.4.3.2 Closing back cross aisle

This scenario investigates the effects of closing the back aisle, which limits the available paths through the warehouse. This situation shows what happens if the back aisle is used for the storage of bulk products. Currently, part of the back aisle is already used for this, but there is still enough space for an order picker to use the back aisle to cross from one aisle to another. This scenario investigates how the batching and routing algorithms perform when the back aisle is not available for crossing between picking aisles anymore. For this, three experiments are conducted. Exp. 0 is the baseline, examining the situation with an open and usable back aisle. In Exp. 1, the back cross aisle is closed but the batches evaluated are originally created for a warehouse structure in which this back aisle is open. In Exp. 2, new batches are created, based on a warehouse where the back aisle is closed. By comparing the results of the three experiments, summarised in Table 5.16, the effects of the back cross aisle on the performance of the solution approach can be evaluated.

Closing the back aisle while using the original batches (Exp. 1) leads to higher distances and picking times. This is expected, since taking away a pathway reduces the route flexibility, requiring more detours to access all pick locations in a batch. The average running times change only slightly when compared to the baseline situation (Exp. 0) as the routing algorithm processes the same batches.

When new batches are formed under the closed back aisle situation (Exp. 2), the impact is more dataset dependent. In D2, the number of batches increase from 5 to 6, with the travel distance and average picking time increasing when compared to both the baseline and Exp. 1. This suggests that the batching algorithm creates smaller batches that require more routes, which leads to higher travel distances. In D3, the newly formed batches perform better than the original batches under the new warehouse structure. Interestingly, while the total distance of the newly created batches is larger than the baseline situation (Exp. 0), the average total picking time is smaller, showing that unique pick locations are better grouped together. In D4 and D5, the new batches yield the best results across all three experiments. This indicates that the algorithm adapts well in these datasets. Additionally, the running times are reduced quite significantly for both datasets, which indicates that the newly formed batches are easier to optimise.

Table 5.16 Results of closing back cross aisle

Dataset	*Experiment*	*Avg batching time*	*Number of batches*	*Avg distance*	*Avg picking time*	*Avg running time*
D2	Exp. 0: Baseline	522.79	5	960.8	2233.46	364.86
	Exp. 1: Original batch	522.79	5	1002.0	2270.91	397.54
	Exp. 2: New batch	470.21	6	1106.0	2390.45	353.19
D3	Exp. 0: Baseline	534.20	6	859.2	4011.09	531.46
	Exp. 1: Original batch	534.20	6	961.4	4104.00	523.29
	Exp. 2: New batch	524.73	6	881.0	3965.91	476.31
D4	Exp. 0: Baseline	389.48	5	1117.6	2258.00	480.83
	Exp. 1: Original batch	389.48	5	1179.4	2314.18	464.19
	Exp. 2: New batch	342.55	5	1103.4	2225.09	363.77
D5	Exp. 0: Baseline	600.79	8	1239.6	3662.91	290.57
	Exp. 1: Original batch	600.79	8	1342.2	3756.18	304.44
	Exp. 2: New batch	604.18	8	1149.0	3520.55	159.94

Overall, these results confirm that closing the cross aisle generally reduces the routing efficiency, although the extent of this impact depends largely on the structure of the dataset. In some cases (D4 and D5), newly formed batches based on the new warehouse structure surpass the baseline in both the travel distance and picking times, suggesting that the batching algorithm can adapt effectively under the new constraints. However, in other datasets (D2 and D3), the new batches perform worse than the baseline, showing the importance of maintaining travel paths open for the order pickers.

5.5 Conclusion

This chapter answers the research question: *"How does the developed solution improve warehouse efficiency compared to the current situation?"*. The outlined experimental framework enabled the selection and tuning of key algorithmic components such as the batch formation method, operator selection strategy, and simulated annealing parameters. The results show that the Clarke & Wright savings algorithm, in combination with a random operator strategy and carefully selected SA parameters, yield high performing and robust solutions.

After the algorithms were tuned, comparisons were made between the metaheuristic approach and the MILP model. These comparisons demonstrate that the simulated annealing algorithm produces solutions of similar quality in a fraction of the computation time. Then, the routes were benchmarked against the company's current practice, which revealed that the potential improvements of the algorithm are substantial, both in terms of travel distance and picking time. By grouping the orders efficiently and creating optimised routes, it was shown that the travel distance could consistently be decreased with more than 50% compared to the manual batching and routing practices.

Further scenario analyses highlighted the flexibility of the solution under varying operational conditions. Here, the effects of different batch sizes, technological implementations and changes in warehouse layouts were examined and reported. The results show that the algorithm can adapt to new requirements, providing high performing solutions.

The developed solution approach offers significant operational benefits, with reductions in travel distance and picking time, by intelligently forming batches and optimising routes which reduce unproductive travel and manual inefficiencies. The performance of the algorithms is robust and scalable, making them practical and valuable for Company X.

6. Conclusions and recommendations

This chapter summarises the research and its results in Section 6.1. After the conclusion of the research is given, recommendations for Company X are presented in Section 6.2. Section 6.3 discusses the limitations and actionable points for future research. The theoretical and practical contributions of this research are discussed in Section 6.4.

6.1 Conclusion

This study started with an investigation of the different operational processes at Company X, looking for potential reasons of the experienced inefficiencies at the fulfilment centre. After the core problem was identified, research questions were formulated to guide the research towards answering the main research question, which states:

“How can Company X align its order picking processes with industry best practices to improve efficiency?”

To answer this question, a comprehensive analysis of the current warehouse operations was performed where the key processes and sources of inefficiencies were identified. Based on the gained insights, the problem was formulated as a two-stage optimisation problem, addressing both the batch formation and picker route optimisation.

To solve the problem, a two-stage solution approach was developed. For the first stage, a batch formation algorithm was developed to group customer orders into dense, efficient batches. It was found that the Clarke & Wright savings algorithm achieved the best results. For the second stage, a route optimisation algorithm was implemented to find the best route along all pick locations within a batch. An exact Mixed Integer Linear Programming (MILP) model was created to act as a benchmark for the more scalable Simulated Annealing (SA) algorithm. Within the SA framework, a random operator selection strategy yielded the best balance between solution quality and operating speed. After tuning the parameters of the SA, a high-performing configuration was found, finishing the setup of the algorithm.

By benchmarking the solution approach to the MILP and historical data of Company X, it was found that the SA model provided significant improvements over the current situation. The total number of batches required to fulfil all orders in a given dataset was reduced significantly, which resulted in a reduction of the travel distance of more than 50%. The scenario analysis demonstrated that batch size, technological innovations, and warehouse layout changes can have some significant impact on the total travel distance and picking time.

The results show that the algorithm provides high quality solutions. However, to ensure that the solution model has long lasting impact on the day-to-day operations of Company X, it has to be implemented and integrated correctly in the workflow of the order pickers. This research finds that the current situation would not sufficiently improve if the generated optimal batches and routes are simply provided to the order pickers on paper, as currently is the situation. In such cases, pickers may not follow the provided optimal route exactly, since they tend to make their own choices after visually scanning the entire batch list and required pick locations. To realise the true gains of the developed solution, it must be ensured that order pickers are guided through the optimal route step by step, only revealing the next location once the current pick is finished. For this, technological innovations such as handheld scanners, or similar digital picking support systems, are essential for this and ensure that the optimised routes are actually followed on the warehouse floor. So, to achieve substantial improvements in the order picking process of Company X, it is critical to combine the designed solution approach with supportive picking technologies.

6.2 Recommendations

Based on the outcomes of this research, the following recommendations are proposed to Company X:

- **Implement the solution approach:** Adopt the proposed solution approach within the current WMS, to ensure that the batches and routes are optimised. The algorithms are effective in reducing the number of batches, travel time and picking times, resulting in substantial improvements over the current way of working.
- **Invest in digital picking technologies:** To realise the full benefits of the optimisation algorithms, order pickers should be equipped with handheld barcode scanners or similar digital picking devices. These technologies enable real-time route guidance to ensure that the optimised routes are followed by the order pickers by eliminating deviations from the suggested route. It is advisable to coordinate the choice of picking device with the moderators of the WMS to ensure system compatibility and seamless integration.
- **Record product volumes and cart capacities:** To improve batch accuracy, Company X should measure and record the volume of all their products and the capacity of the picking carts. This will allow the batch formation algorithm to form batches based on the physical space and weight of all the products within a batch, rather than on a maximum number of pick locations or maximum quantity of items. Basing batch decisions on the item characteristics and cart limitations, operational efficiency and picker ergonomics can be improved further.
- **Plan for integration and staff training:** Once a suitable picking technology is chosen, ensure that the technology and solution approach are integrated gradually, so that all employees can get used to the new way of working. To help with this, targeted training and clear communication could help to show order pickers and supervisors how the new system works and what its benefits are. By keeping the staff involved in the integration period and incorporating their feedback, the chance of a successful adoption is increased and the resistance to change is reduced.

6.3 Limitations and future research

This study encounters a several limitations that need to be mentioned. First, the experiments and parameter tuning are performed on only two datasets. While these datasets are representative of the operational peak of regular days, a broader evaluation across more days with varying demand profiles would further strengthen the robustness and generalisability of the findings.

Second, a limitation is the absence of product volume and weight data. As described in the recommendations, this data may lead to more accurate batching decisions, especially for large or irregular sized products. The current use of a maximum for the number of pick locations and quantity of items limits the creation of batches based on volume and cart capacity.

Third, challenges could arise with the computational aspects of the batch formation algorithm when instance sizes grow. As observed in the scalability experiments, the batch formation time increases significantly for larger datasets, potentially impacting real-time applicability. Future research could explore strategies for parallel computation of batches.

Another aspect that could be looked at in the future is the addition of time windows for orders, so that orders with an early courier time are batched before orders that are not close to their courier pickup time. This could potentially help reducing the batch formation time as well, since two different groups of orders can be made, one group of orders that are close to being picked up and one with time orders that are not close to their deadline yet. This separation of orders results in smaller groups of orders, which can potentially be optimised quicker. Future work can investigate whether adding time windows to orders have significant impact on the batch creation time.

By addressing these limitations, future studies can enhance the efficiency, practical relevance and robustness of the proposed solution approach.

6.4 Contribution

This thesis makes several contributions to both theory and practice in the field of order picking optimisation within warehouses.

Contribution to theory

This research contributes to the existing literature by the explicit modelling and analysis of ultra-narrow aisles within the context of a real-world warehouse. Overall, there is limited research on the implementation of this type of aisles, but all the research that is available considers warehouse layouts that only contain the ultra-narrow aisles. Since this study models the real-world layout of Company X, it had to combine the much more complicated and realistic situation of ultra-narrow aisles with standard aisles, something that has not been found in literature before. This combination of aisle types introduced unique routing and batching challenges, such as same side entry and exit side for only a specific area of the warehouse. By developing a solution approach for this mixed layout, this thesis provides new insights and lays a foundation for future research within this area.

Furthermore, this study extends the application of savings algorithms to batching with real operational constraints. By combining this with the simulated annealing metaheuristic for route optimisation, a robust algorithm is created that is used for the evaluation of different scenarios, providing empirical insights into the effect of key parameters and warehouse layouts. These findings help to bridge the gap between theoretical optimisation research and real-world warehouse challenges.

Contribution to practice

In practical terms, this study delivers a robust, high performing solution approach that can be implemented at Company X. The algorithms have shown to reduce the number of batches required to fulfil all orders, while also reducing the travel distance and picking times. As the model is benchmarked against both the current operational practice and the MILP model, it shows that the metaheuristic is effective and provides real-world value.

Importantly, this research highlights the need to combine the solution approach with the implementation of picking support systems, to ensure that the optimised routes are actually followed by the order pickers.

Bibliography

Aerts, B., Cornelissens, T., & Sörensen, K. (2021). The joint order batching and picker routing problem: Modelled and solved as a clustered vehicle routing problem. *Computers & Operations Research*, *129*, 105168. https://doi.org/10.1016/j.cor.2020.105168

Bartholdi, J. J., & Hackman, S. T. (2019). *Warehouse & Distribution Science* (0.981). https://www.warehouse-science.com/

Battini, D., Calzavara, M., Persona, A., & Sgarbossa, F. (2015). A comparative analysis of different paperless picking systems. *Industrial Management & Data Systems*, *115*(3), 483–503. https://doi.org/10.1108/imds-10-2014-0314

Boysen, N., Briskorn, D., Fedtke, S., & Schmickerath, M. (2019). Automated sortation conveyors: A survey from an operational research perspective. *European Journal of Operational Research*, *276*(3), 796–815. https://doi.org/10.1016/j.ejor.2018.08.014

Boysen, N., Briskorn, D., Füßler, D., & Stephan, K. (2023). Put it in the bag: Order fulfillment with a pocket sorter system. *Naval Research Logistics (NRL)*, *70*(8), 858–877. https://doi.org/10.1002/nav.22137

Boysen, N., & De Koster, R. (2025). 50 years of warehousing research—An operations research perspective. *European Journal of Operational Research*, *320*(3), 449–464. https://doi.org/10.1016/j.ejor.2024.03.026

Boysen, N., De Koster, R., & Weidinger, F. (2019). Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*, *277*(2), 396–411. https://doi.org/10.1016/j.ejor.2018.08.023

Casella, G., Volpi, A., Montanari, R., Tebaldi, L., & Bottani, E. (2023). Trends in order picking: a 2007–2022 review of the literature. *Production & Manufacturing Research*, *11*(1). https://doi.org/10.1080/21693277.2023.2191115

Chabot, T., Lahyani, R., Coelho, L. C., & Renaud, J. (2016). Order picking problems under weight, fragility and category constraints. *International Journal of Production Research*, *55*(21), 6361–6379. https://doi.org/10.1080/00207543.2016.1251625

Chen, F., Wang, H., Qi, C., & Xie, Y. (2013). An ant colony optimization routing algorithm for two order pickers with congestion consideration. *Computers & Industrial Engineering*, *66*(1), 77–85. https://doi.org/10.1016/j.cie.2013.06.013

Clarke, G., & Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Operations Research, 12(4), 568–581. https://doi.org/10.1287/opre.12.4.568

De Koster, R. (2022). Warehousing 2030. In *Springer eBooks* (pp. 243–260). https://doi.org/10.1007/978-3-030-95764-3_14

De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, *182*(2), 481–501. https://doi.org/10.1016/j.ejor.2006.07.009

De Koster, R., & Van Der Poort, E. (1998). Routing orderpickers in a warehouse: A comparison between optimal and heuristic solutions. *IIE Transactions*, *30*(5), 469–480. https://doi.org/10.1023/a:1007599307171

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

Dukic, G., & Oluic, C. (2007). Order-picking methods: improving order-picking efficiency. International Journal of Logistics Systems and Management, 3(4), 451. https://doi.org/10.1504/ijlsm.2007.013214

Elsayed, E. A., & Unal, O. I. (1989). Order batching algorithms and travel-time estimation for automated storage/retrieval systems. International Journal of Production Research, 27(7), 1097–1114. https://doi.org/10.1080/00207548908942610

Fragapane, G., De Koster, R., Sgarbossa, F., & Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*, *294*(2), 405–426. https://doi.org/10.1016/j.ejor.2021.01.019

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

Haase, J., & Beimborn, D. (2017). Acceptance of Warehouse Picking Systems. *Proceedings of the 2017 ACM SIGMIS Conference on Computers and People Research*, 53–60. https://doi.org/10.1145/3084381.3084409

Heerkens, H., & Van Winden, A. (2017). *Solving Managerial Problems Systematically*. Noordhoff Uitgevers.

Hompel, M. T., & Schmidt, T. (2007). Warehouse management : automation and organisation of warehouse and order picking systems. In *Springer eBooks*. http://dx.doi.org/10.1007/978-3-540-35220-4

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

Ilanković, N., Živanić, D., & Zelić, A. (2020). Augmented Reality in Order-picking processes – Advantages and Disadvantages. *Logisztika – Informatika – Menedzsment*, *5*(1), 4–12. https://doi.org/10.29177/lim.2020.1.4

Kapoor, G., Lee, Y., Sikora, R., & Piramuthu, S. (2024). Drone-based warehouse inventory management of perishables. *International Journal of Production Economics*, 109437. https://doi.org/10.1016/j.ijpe.2024.109437

Kulak, O., Sahin, Y., & Taner, M. E. (2011). Joint order batching and picker routing in single and multiple-cross-aisle warehouses using cluster-based tabu search algorithms. *Flexible Services and Manufacturing Journal*, *24*(1), 52–80. https://doi.org/10.1007/s10696-011-9101-8

Kumar, D., Singh, R. K., Mishra, R., & Wamba, S. F. (2022). Applications of the internet of things for optimizing warehousing and logistics operations: A systematic literature review and future research directions. *Computers & Industrial Engineering*, *171*, 108455. https://doi.org/10.1016/j.cie.2022.108455

Liu, W., Lin, C., Chiu, C., Tsao, Y., & Wang, Q. (2014). Minimizing the Carbon Footprint for the Time-Dependent Heterogeneous-Fleet Vehicle Routing Problem with Alternative Paths. *Sustainability*, *6*(7), 4658–4684. https://doi.org/10.3390/su6074658

Lu, W., McFarlane, D., Giannikas, V., & Zhang, Q. (2016). An algorithm for dynamic order-picking in warehouse operations. *European Journal of Operational Research*, *248*(1), 107–122. https://doi.org/10.1016/j.ejor.2015.06.074

Malang, C., Charoenkwan, P., & Wudhikarn, R. (2023). Implementation and Critical factors of Unmanned Aerial Vehicle (UAV) in Warehouse Management: A systematic literature review. *Drones*, *7*(2), 80. https://doi.org/10.3390/drones7020080

Masae, M., Glock, C. H., & Grosse, E. H. (2020). Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*, *224*, 107564. https://doi.org/10.1016/j.ijpe.2019.107564

Nathania, F., & Komarudin, K. (2023). An Efficient Simulated Annealing Algorithm for the Joint Order Batching and Picker Routing Problem in Manual Order Picking Systems. In *8th North America Conference on Industrial Engineering and Operations Management*. https://doi.org/10.46254/na8.20230142

Nishar, S. (2024). Intelligent Decision-Making in Warehouse Management: How AI automation improves inventory tracking, order fulfillment, and logistics efficiency compared to drone technology. *Intelligent Control and Automation*, *15*(01), 1–8. https://doi.org/10.4236/ica.2024.151001

Pansart, L., Catusse, N., & Cambazard, H. (2018). Exact algorithms for the order picking problem. *Computers & Operations Research*, *100*, 117–127. https://doi.org/10.1016/j.cor.2018.07.002

Ratliff, H. D., & Rosenthal, A. S. (1983). Order-Picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Operations Research*, *31*(3), 507–521. https://doi.org/10.1287/opre.31.3.507

Richards, G. (2011). *Warehouse Management: A Complete Guide to Improving Efficiency and Minimizing Costs in the Modern Warehouse*. Kogan Page Publishers.

Roodbergen, K. J. (2001). *Layout and routing methods for warehouses*. https://ci.nii.ac.jp/naid/20000715707

Roodbergen, K. J., & De Koster, R. (2001). Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research*, *133*(1), 32–43. https://doi.org/10.1016/s0377-2217(00)00177-6

Scholz, A., Henn, S., Stuhlmann, M., & Wäscher, G. (2016). A new mathematical programming formulation for the Single-Picker Routing Problem. *European Journal of Operational Research*, *253*(1), 68–84. https://doi.org/10.1016/j.ejor.2016.02.018

Schrotenboer, A. H., Wruck, S., Roodbergen, K. J., Veenstra, M., & Dijkstra, A. S. (2017). Order picker routing with product returns and interaction delays. *International Journal of*

Production Research, *55*(21), 6394–6406. https://doi.org/10.1080/00207543.2016.1206982

Shetty, N., Sah, B., & Chung, S. H. (2020). Route optimization for warehouse order picking operations via vehicle routing and simulation. *SN Applied Sciences*, *2*(2). https://doi.org/10.1007/s42452-020-2076-x

Sodiya, N. E. O., Umoga, N. U. J., Amoo, N. O. O., & Atadoga, N. A. (2024). AI-driven warehouse automation: A comprehensive review of systems. *GSC Advanced Research and Reviews*, *18*(2), 272–282. https://doi.org/10.30574/gscarr.2024.18.2.0063

Stević, Ž., Zavadskas, E. K., Tawfiq, F. M. O., Tchier, F., & Davidov, T. (2022). Fuzzy Multicriteria Decision-Making model based on Z numbers for the evaluation of information technology for order picking in warehouses. *Applied Sciences*, *12*(24), 12533. https://doi.org/10.3390/app122412533

Theys, C., Bräysy, O., Dullaert, W., & Raa, B. (2010). Using a TSP heuristic for routing order pickers in warehouses. *European Journal of Operational Research*, *200*(3), 755–763. https://doi.org/10.1016/j.ejor.2009.01.036

Tsai, C., Liou, J. J. H., & Huang, T. (2008). Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time. *International Journal of Production Research*, *46*(22), 6533–6555. https://doi.org/10.1080/00207540701441947

Zulj, I., Glock, C. H., Grosse, E. H., & Schneider, M. (2018). Picker routing and storage-assignment strategies for precedence-constrained order picking. *Computers & Industrial Engineering*, *123*, 338–347. https://doi.org/10.1016/j.cie.2018.06.015

Žulj, I., Kramer, S., & Schneider, M. (2018). A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem. European Journal of Operational Research, 264(2), 653–664. https://doi.org/10.1016/j.ejor.2017.06.056

Appendices

A. Pick list

B2025-08412

Amount of picklists: **17**

Amount of products: **23**

	Product	Locatie	Amount in picklists
1x	Product A	Deck.02.07.B	1x P2025-168969
3x	Product B	Deck.02.12.B	3x P2025-168977
1x	Product C	Deck.02.16.B	1x P2025-168977
3x	Product D	Grof	1x P2025-168974 1x P2025-168975 1x P2025-168976
1x	Product E	Pallet.01.18	1x P2025-168977
1x	Product F	Pallet.01.21	1x P2025-168969
1x	Product G	Pallet.01.31	1x P2025-168969
1x	Product H	Pallet.02.06	1x P2025-168978
1x	Product I	Pallet.02.16	1x P2025-168969
1x	Product J	Pallet.04.14	1x P2025-168971
1x	Product K	Pallet.04.31	1x P2025-168980
1x	Product L	Pallet.05.23	1x P2025-168981
1x	Product M	Pallet.06.14	1x P2025-168976
3x	Product N	Pallet.09.07	3x P2025-168972
1x	Product O	Shelf B.11.14.2.C	1x P2025-168982
1x	Product P	Shelf B.11.14.3.A	1x P2025-168985
1x	Product Q	Shelf B.7.1.3.C	1x P2025-168968
1x	Product R	Shelf B.7.11.4.A	1x P2025-168970
3x	Product S	Shelf B.7.13.1.B	3x P2025-168979
1x	Product T	Shelf B.7.3.4.C	1x P2025-168983
1x	Product U	Shelf.14.12.3.B	1x P2025-168984
1x	Product V	Shelf.14.13.2.A	1x P2025-168984
1x	Product W	Shelf.14.14.4.C	1x P2025-168982

Figure A.1 Example of a batch list

B. Results SA vs MILP

*Table B.1 Detailed results for all batches of D1 for the SA and MILP models. A * indicates that the MILP did not find feasible solutions for all batches, so the shown distance is a sum over batches with solutions only.*

Method	Metric	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	Total
SA	Distance (m)	83	83	83	82	83	88	223	350	105	290	239	166	72	1947
	Run time (s)	10.22	9.82	16.13	17.34	14.84	7.40	181.95	43.06	4.98	48.58	197.06	108.51	136.71	796.60
MILP (60 s)	Distance (m)	83	83	83	-	83	88	255	-	105	-	238	-	-	1018*
	Gap (%)	0.0	0.0	0.0	-	0.0	0.0	0.0	-	0.0	-	0.0	-	-	-
	Run time (s)	4.92	2.57	3.16	122.85	3.14	2.73	69.87	132.61	9.11	129.77	75.06	125.68	124.26	805.74
MILP (1800 s)	Distance (m)	83	83	83	-	83	88	223	350	105	290	238	222	72	1920*
	Gap (%)	0.0	0.0	0.0	-	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
	Run time (s)	5.46	3.53	4.06	3605.32	4.79	2.86	285.21	359.71	7.61	1015.51	1817.72	2281.25	2009.35	11402.38
	# UNAs	0	0	0	0	0	0	3	2	0	2	4	3	4	

Table B.2 Detailed results for all batches of D2 for the SA and MILP models

Method	Metric	B1	B2	B3	B4	B5	Total
SA	Distance (m)	145	194	196	328	83	946
	Run time (s)	42.20	71.67	30.64	202.06	11.19	357.76
MILP (60 s)	Distance (m)	145	193	196	385	83	1002
	Gap (%)	0.0	0.0	0.0	0.0	0.0	-
	Run time (s)	80.87	9.54	39.40	74.03	3.26	207.094
MILP (1800 s)	Distance (m)	145	193	196	385	83	1002
	Gap (%)	0.0	0.0	0.0	0.0	0.0	-
	Run time (s)	79.08	7.67	35.58	1814.54	3.02	1939.88
	# UNAs	3	3	2	3	0	

Table B.3 Detailed results for all batches of D3 for the SA and MILP models.

Method	Metric	B1	B2	B3	B4	B5	B6	Total
SA	Distance (m)	88	131	218	245	92	83	857
	Run time (s)	16.78	134.17	327.60	15.21	19.15	11.80	524.71
MILP (60 s)	Distance (m)	88	131	217	245	92	83	856
	Gap (%)	0.0	0.0	0.0	0.0	0.0	0.0	-
	Run time (s)	3.19	58.03	19.02	32.80	93.52	3.35	209.91
MILP (1800 s)	Distance (m)	88	131	217	245	92	83	856
	Gap (%)	0.0	0.0	0.0	0.0	0.0	0.0	-
	Run time (s)	3.46	49.82	14.99	28.95	1846.17	3.40	1946.80
	# UNAs	0	4	4	1	2	0	

Table B.4 Detailed results for all batches of D4 for the SA and MILP models.

Method	Metric	B1	B2	B3	B4	B5	Total
SA	Distance (m)	291	107	283	331	95	1107
	Run time (s)	173.00	9.61	119.77	133.76	50.81	486.96
MILP (60 s)	Distance (m)	339	105	367	385	95	1291
	Gap (%)	0.0	0.0	0.0	0.0	0.0	-
	Run time (s)	12.19	63.15	73.64	73.62	2.78	225.37
MILP (1800 s)	Distance (m)	339	105	367	331	95	1237
	Gap (%)	0.0	0.0	0.0	0.0	0.0	-
	Run time (s)	13.88	66.87	1818.62	495.23	3.85	2398.45
	# UNAs	4	0	3	4	4	

*Table B.5 Detailed results for all batches of D5 for the SA and MILP models. A * indicates that the MILP did not find feasible solutions for all batches, so the shown distance is a sum over batches with solutions only.*

Method	Metric	B1	B2	B3	B4	B5	B6	B7	B8	Total
SA	Distance (m)	86	358	183	122	83	162	78	137	1209
	Run time (s)	11.04	74.31	169.91	17.57	12.26	13.36	7.66	11.96	318.07
MILP (60 s)	Distance (m)	86	358	-	122	78	212	78	137	1071*
	Gap (%)	0.0	0.0	-	0.0	0.0	0.0	0.0	0.0	-
	Run time (s)	2.85	16.45	125.14	5.57	12.58	64.76	2.95	10.90	241.20
MILP (1800 s)	Distance (m)	86	358	183	122	78	162	78	137	1204
	Gap (%)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-
	Run time (s)	3.06	18.69	1973.42	6.44	14.01	221.80	3.20	10.46	2251.05
	# UNAs	0	3	4	2	0	1	0	1	