UNIVERSITY OF TWENTE.

Faculty of Industrial Engineering & Management

Improving storage efficiency by implementing an optimized box allocation method for the Kardex shuttles at Benchmark

Name: Thom Baas

First Supervisor: Lin Xie
Second Supervisor: Ipek Topan

Company Supervisor 1: Herman Wind
Company Supervisor 2: Lars Tijhuis

Benchmark

Management Summary

Motivation

This research was conducted for Benchmark Electronics to investigate the impact of a box allocation method on the storage efficiency of their Kardex shuttles. In the warehouse of Benchmark, ten Kardex shuttles are located which store SKUs in six different box types. Benchmark addresses that frequently high occupancy rates are faced for these box types, which results in several box types having no free boxes left to store supplied SKUs at certain periods. Benchmark addresses that implementing a box allocation method might prevent this undesired situation where there is no capacity for certain box types. A box allocation method could determine the recommended box type for supplied SKUs and propose changes to box types for boxes currently placed in the Kardex shuttles. The purpose would be to recommend boxes and changes that decrease the volume required to store all SKUs in the Kardex shuttles, increasing the storage efficiency. Therefore, this research was conducted to provide clarity on whether a box allocation method could improve the storage efficiency in the Kardex shuttles.

Research Question

The main research question for this research is formulated as follows:

"How can more storage efficiency be obtained by using an optimized box allocation method for the Kardex shuttles at Benchmark? "

The developed box allocation method consists of the box division model and reallocation model. The box division model determines the recommended box type for supplied SKUs and proposes reallocations for boxes stored in the Kardex shuttles. It has three options for improving storage efficiency that decrease the occupied volume of boxes stored on the Kardex shuttles. These options are OTO reallocations, OTM reallocations, and carrier type changes.

- OTO Reallocations: The box division model proposes several One-to-One (OTO) reallocations that lead to instant volume savings in the Kardex shuttles. It changes a large box type by one box of a smaller box type to achieve volume savings in the Kardex shuttles.
- OTM Reallocations: One-to-Many (OTM) reallocations are proposed by the box division model if volume can be saved by changing one box to multiple other boxes of a different box type for boxes stored in the Kardex shuttles. This reallocation method differs from OTO reallocations since multiple boxes are involved for the recommended box type in the OTM reallocation.
- Carrier Type Changes: Another reallocation method proposed by the box division model is carrier type changes. Carrier type changes change the carrier types of carriers, resulting in the placement of boxes of a different box type on carriers selected for carrier type changes. This is useful if a high occupancy rate for a certain box type in the Kardex shuttles is faced. The box division model uses a desired maximum occupancy rate per box type and proposes carrier type changes to achieve this desired maximum occupancy rate. Benchmark mentioned that an occupancy rate of 90% would be desired for the box division model.

The box division model proposes reallocations based on these three reallocation methods and calculates the total costs of purchasing materials for these operations e.g. additional boxes. The reallocation model is developed to calculate the completion time and total labour costs of executing the reallocations proposed by the box division model. The reallocation model uses several steps to schedule the execution of the reallocations which all result in labour costs for each step.

Several KPIs are created for Benchmark to see the impact on storage efficiency due to the implementation of the box allocation method. The first one is *TotalStorageVolume* which shows the total volume available for storage in the Kardex shuttles. The second one is *TotalOccupiedVolume* which shows the storage volume that is occupied in the Kardex shuttles. The third KPI is

TotalUnoccupiedVolume which indicates the unused volume in the Kardex shuttles and could be used for placing additional carriers and boxes. Using these KPIs provides insights into the volume usage of the Kardex shuttles.

Results

The box allocation method significantly improves the storage efficiency of the Kardex shuttles. Table 1 shows a comparison between the current situation and the improved situation after the box allocation method is used.

Table 1: Comparison between the situation before and after implementing the box allocation method

KPI	Current Situation	Improved Situation
TotalOccupiedVolume	288.1 m3	275.5 m3
TotalUnoccupiedVolume	38.9 m3	51.5 m3

Using the reallocations provided by the box division model, Benchmark can achieve a *TotalUnoccupiedVolume* of 51.5 m3 which is an improvement over the 38.9 m3 from the current situation. Hence, a decrease in the *TotalOccupiedVolume* can be seen compared to the current situation. Note that this is based on a maximum occupancy rate of 90% as desired by Benchmark and no box types exceed this value. The material costs of these reallocations are determined at € 2131.46.

The unoccupied volume can be used by Benchmark for the placement of additional carriers as shown in Table 2. Table 2 provides the volume required to place additional carriers of a carrier type in the Kardex shuttle and the boxes placed on each carrier type. Based on the *TotalUnoccupiedVolume* a combination of carriers to place in the Kardex shuttles and the number of boxes that would be added for each box type can be determined.

Table 2: Required volume and added boxes by placement of a carrier type

CarrierType	Required Volume	Boxes on Carrier Type
B001	0.388 m3	80 B001 boxes
B002	0.518 m3	40 B002 boxes
B003	0.647 m3	20 B003 boxes
B004	0.906 m3	10 B004 boxes
B005	0.388 m3	10 B005 boxes
B006	0.518 m3	34 B006 & 11 B002 boxes

The reallocation model provided the completion time and total labour costs of executing the reallocations. It uses several different steps which all include labour costs. Table 3 shows an overview of the total costs for implementing the box allocation method at Benchmark. It shows both the material costs and labour costs split up over the different steps in the reallocation model.

Table 3: Total costs of implementing the box allocation method at Benchmark

Operation	Cost
Material Costs	€ 2131.46
Extracting Reallocations	€ 335.09
Carrier Type Changes	€ 259.92
Execute Extracted OTM Reallocations	€ 497.84
Execute Extracted OTO Reallocations	€ 563.15
OTM Reallocations	€ 1880.42
OTO Reallocations	€ 2088.60
Total Cost	€ 7756.48

Conclusion

The results of this research showed that Benchmark could significantly profit from implementing a box allocation method for the Kardex shuttles. The box allocation method successfully decreased the *TotalOccupiedVolume* while increasing the *TotalUnoccupiedVolume*. The *TotalOccupiedVolume* decreased while maintaining an occupancy rate below 90% for all box types. This shows that less space is required to store the SKUs in the Kardex shuttles while maintaining a balanced occupancy rate under 90% for all box types.

The *TotalUnoccupiedVolume* is determined at 38.9 m3 in the current situation. After performing an operation that adjusts the carrier heights, this unoccupied volume could be used by Benchmark for placing additional carriers and boxes in the Kardex shuttles. This means there is already space for storage expansion in the Kardex shuttles without implementing the box allocation method. In case Benchmark implements the box allocation method, an even larger *TotalUnoccupiedVolume* of 51.5 m3 can be realized. The total costs of implementing the box allocation method are € 7756.48. Hence, a significant amount of storage capacity can be added to the Kardex shuttle by Benchmark at a relatively small investment compared to an additional Kardex shuttle that costs approximately € 75.000.

Preface

Dear Reader,

In front of you lays my MSc Thesis ”Improving storage efficiency by implementing optimized box allocation for the Kardex shuttles at Benchmark” which I created to achieve the Master Industrial Engineering & Management. This MSc Thesis was performed at Benchmark Electronics located in Almelo. I want to thank my company supervisors Herman Wind and Lars Tijhuis for the nice collaboration and valuable discussions we had. I also want to thank my first supervisor Lin Xie and second supervisor Ipek Topan for supporting me with valuable feedback throughout this journey. All in all, I am looking back at a nice period in which I managed to develop new skills that I will take on in my professional career.

Enjoy reading my MSc Thesis.

Thom Baas

Contents

Table of Figures

List of Abbreviations & Terms

Abbreviations & Terms	Definition
Reallocations	Changes to the box division in the Kardex shuttles
One-To-One Reallocation (OTO)	Put the contents of one box into a single other box
One-To-Many Reallocation (OTM)	Put the contents of one box into multiple other boxes
(Kardex) Shuttle	Kardex shuttle is a VLM placed in the Warehouse
Carrier	Carriers are placed in a Kardex shuttle to store boxes of a certain type
VLM	Vertical Lift Machine
PPG	PowerPick Global
LN	Infor LN ERP System
SKU	Stock Keeping Unit
SMD	Surface Mounted Device
PCBA	Printed Circuit Board Assembly

Chapter 1- Introduction

In this chapter, an introduction is given to the thesis. Section 1.1 gives insight into the company Benchmark Electronics. Section 1.2 clarifies the problem context. Section 1.3 formulates the problem statement for this research. Section 1.4 elaborates on the research design set up for this research.

1.1 Background Information

Benchmark Electronics is a high-tech electronics company that operates in multiple countries around the world. The company currently has 23 locations worldwide as shown in Figure 1. One of their locations is based in Almelo, which has been growing quickly over the last few years and has around 800 employees. Benchmark Almelo is focused on doing both Design Engineering and Manufacturing and is also doing design-to-manufacture projects for customers. This is unique in comparison to other Benchmark locations which mainly focus on either Design Engineering or Manufacturing. Due to this rapid growth, the pressure on the warehouse is increasing. Benchmark is investing in more storage racks, but this cannot be done indefinitely since space is limited. Therefore, Benchmark wants to use the current storage space as efficiently as possible.

Figure 1: Map visualizing locations of Benchmark Electronics around the World

1.2 Problem Context

This section discusses the problem context within Benchmark. It will provide insights into several important subjects surrounding the processes occurring at the warehouse of Benchmark.

1.2.1 Kardex Shuttles

Benchmark has invested in the storage expansion of their warehouse by obtaining two additional Kardex shuttles, which are automated storage racks as can be seen in Figure 2. Kardex shuttles are a type of vertical lift machine (VLM) produced by Kardex Remstar. Benchmark currently owns ten Kardex shuttles in which no capacity is left in some periods. Therefore, two additional Kardex shuttles were purchased recently. However, it will become extremely hard to place more Kardex shuttles in the warehouse after the delivery of these additional Kardex shuttles.

In periods when the Kardex shuttles are full, some supplied stock-keeping units (SKUs) are being stored in boxes on pallets in a separate section of the warehouse. This situation where Kardex shuttles have no space left to store boxes should be prevented in the future. Although acquiring two additional Kardex shuttles will increase the storage capacity, Benchmark seeks to optimize the storage efficiency of its current ten Kardex shuttles.

Figure 2: Example of a Kardex shuttle used by Benchmark

1.2.2 Warehouse Layout

Figure 3 shows an illustration of the warehouses at Benchmark. Benchmark makes use of four zones which are the ZKDX1001 (Kardex Zone), General Warehouse, SMD Kardex Storage, and Consigned Storage. Next to these four zones, the incoming goods, outgoing goods, and production areas are visualized. Since this research is aimed at the current ten Kardex shuttles in the ZKDX1001 zone the other zones are discussed briefly but left out of scope in the rest of this thesis.

The ZKDX1001 zone is called the Kardex Zone internally since it contains most of the Kardex shuttles used by Benchmark. Currently, the ZKDX1001 zone contains ten Kardex shuttles named SH01-SH10 respectively. The two additional Kardex shuttles will be placed in the highlighted places in green and will be named SH20 and SH21. The ZKDX1001 zone also contains a space for the storage of bigger goods on pallets within storage racks that are suitable for pallets. Below the ZKDX1001 zone is located the General Warehouse which includes boxes placed on pallets that currently won't fit in the ZKDX1001 zone and other goods.

Near the production area of Benchmark, another Kardex storage zone is located for surface-mounted devices (SMD) is located. This zone is named the SMD Kardex Storage containing SMD rolls and other small items used for printed circuit board assembly (PCBA) at the production location. Lastly, there is a zone dedicated to the in-house storage for a customer of Benchmark. This Consigned Storage zone includes SKUs solely used to produce products for this customer and is separated from all other inventory.

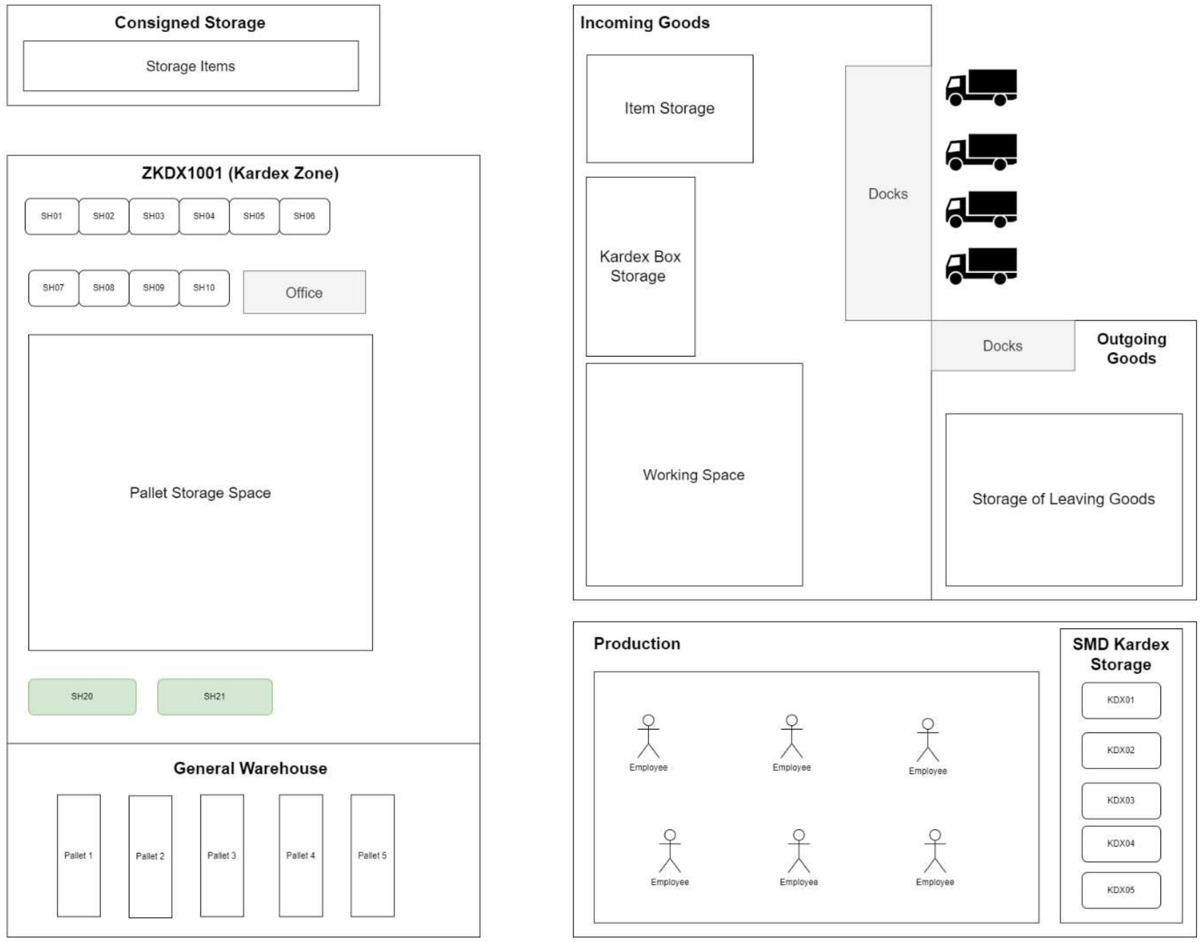

Figure 3: Warehouse Layout at Benchmark

1.2.3 Software Environments

This section explains the software systems used to gain insight into the warehouse and specifically the Kardex shuttles of Benchmark. These three software systems are Infor LN, PowerPick Global, and Rapid Response.

Infor LN

Benchmark uses the ERP software Infor LN containing an extensive collection of data surrounding the warehouse of Benchmark. LN contains valuable information for this research as it includes details about the SKUs. This data can be extracted from LN and used for data analysis throughout this research.

PowerPick Global (PPG)

The Kardex shuttles are making use of the PowerPick Global or PPG software. PPG allows Benchmark to have insight into information surrounding all SKUs stored on the Kardex shuttles. Furthermore, it can be used by employees to automatically find a free storage location in one of the Kardex shuttles for a specific box type. PPG makes use of Infor LN to access and store data on the Kardex shuttles.

Rapid Response

The Rapid Response software is used by Benchmark to generate reports of processes occurring in the warehouse. Rapid Response is connecting to Infor LN to collect data for these reports. There is a wide variety of reports available that can be used to get insight into the operations performed at Benchmark. An example of a frequently used containing information on the demand and supply of SKUs stored at Benchmark.

1.2.4 ZKDX1001 Processes

This section provides an overview of several important policies and activities occurring at the ZKDX1001 zone. These include the storage allocation method, FIFO policy, order picking process, and current way of allocating supplied SKUs to box types.

Storage Allocation Method

Benchmark currently makes use of a random storage policy for the Kardex shuttles within the ZKDX1001 zone. A random storage policy essentially means that stored boxes are not placed at a dedicated storage location but can be put anywhere on the Kardex shuttles. The choice of this random storage policy was based on advice coming from research performed by Jansman (2014). The random allocation is applied via the PPG software using the automatic storage location assignment.

FIFO Policy

In the ZKDX1001 zone, Benchmark uses a FIFO policy for stored SKUs. The FIFO policy makes sure that SKUs with the oldest delivery date are the first to leave in case of demand for the SKUs. This is done to prevent SKUs from exceeding their expiration date. Furthermore, it prevents supplied SKUs from being placed in the same box as a previously supplied identical SKU. This can result in the same SKU being stored in multiple boxes at the Kardex shuttles.

Order Picking

The ZKDX1001 zone contains many SKUs that need to be picked based on demand coming from production. Currently, five order pickers are working full-time to pick up all incoming orders at Benchmark. The order pickers pick orders from Kardex shuttles using a picking list. They perform two tasks simultaneously on two dedicated shuttles to minimize waiting time when retrieving items from the shuttles. After requesting one box to be retrieved from the first Kardex shuttle, they move to the second Kardex shuttle to request another box from this shuttle. After this, the order picker returns to the first shuttle to pick up the requested quantity of an SKU from the retrieved box. Finally, the order picker moves back to the second Kardex shuttle to pick up the requested quantity of the other SKU. This process is repeated by the five order pickers until they have picked all orders on their two dedicated shuttles.

Box Allocation

Employees working at Incoming Goods manage the box allocation of supplied SKUs. The employees decide by their selves which box type is suitable for the supplied quantity of an SKU. At the ZKDX1001 zone, it is observed that doing this results in inefficient storage where e.g. a large B004 box is filled with slow-moving items. This means that it takes a while to empty the large box and over time many lost space is created. One could argue that using multiple smaller boxes would be better since one of the boxes can be removed once it is empty. Therefore, it can be profitable for Benchmark to investigate the impact of determining the recommended box type for SKUs.

1.2.5 SKU Information

Benchmark collects information surrounding SKUs in their information systems. In the warehouse, a selection of this information is printed on a yellow label as shown in Figure 4. These labels are printed when supplied SKUs are being stored at a location in the warehouse by an employee. Important data on this yellow label are the material number, LOT number, warehouse location, and quantity. The material number is used for internal reference of the SKU such that information can be found easily across all information systems at Benchmark. An example of a material number is ASM4022_698_88281-LF, where it can be seen that a material number consists of three parts indicating the production line in the first part. The second and third parts of the material number are related to the specific product and part. Next to the material number, a LOT number is created which is a unique number for each delivered item at Benchmark. The LOT number is important for Benchmark since it keeps track of the delivery date in the second and third parts of the number due to the FIFO policy. This becomes usable when multiple deliveries of the same SKU have taken place and the LOT number can show which one has the earliest delivery date. Figure 4 shows that from the LOT number, it can be observed that this SKU was delivered in week 30 of 2024 due to the second part 2430.

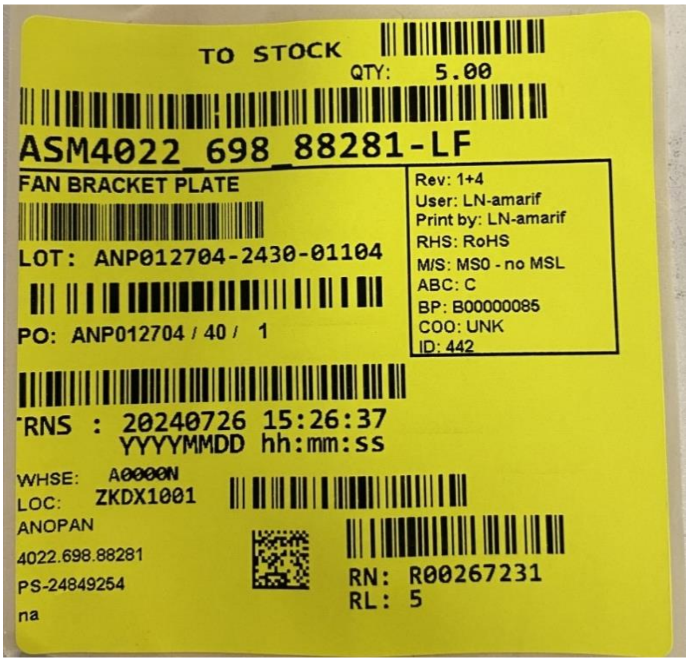

Figure 4: Example of a printed label containing information on the SKU

1.2.6 Storage Process of Incoming Goods

Figure 5 illustrates the complete process of storing an SKU when incoming goods arrive at Benchmark. SKUs arriving at the incoming goods department will be registered and checked before they are stored in the warehouse of Benchmark. Employees working at the incoming goods department can use the information systems of Benchmark to register information surrounding the delivered SKUs. Important data to be registered are the material number, LOT number, and delivered quantity since these will be printed on the yellow label as shown in Figure 4. During the quality check the SKU is inspected on build quality and a check is performed on whether the delivered quantity matches the ordered quantity. In case the quality check is passed the SKU can be stored in the warehouse. If the quality check is not passed the supplier will be contacted and the SKU is not directly stored.

After the SKUs have been checked and registered in the information systems they will be stored in one of the four zones in the warehouse depending on the warehouse location of the SKU. Depending on the warehouse zone the employee performs different actions when storing the SKU. If the SKU is stored at the SMD Kardex Storage or ZKDX1001, the employee first stores the SKU in one or more boxes of a suitable box type based on the experience of the employee. After this, yellow labels are printed for the boxes and a storage location will automatically be assigned in the Kardex shuttles using the PowerPick Global (PPG) software which enables storage of the SKU. As discussed in Section 1.2.3, PPG is the software package delivered by Kardex Remstar for using and analyzing the Kardex shuttles. In case the SKU goes to the General Warehouse or the Consigned Storage it will be placed at a free location again with a yellow label for reference.

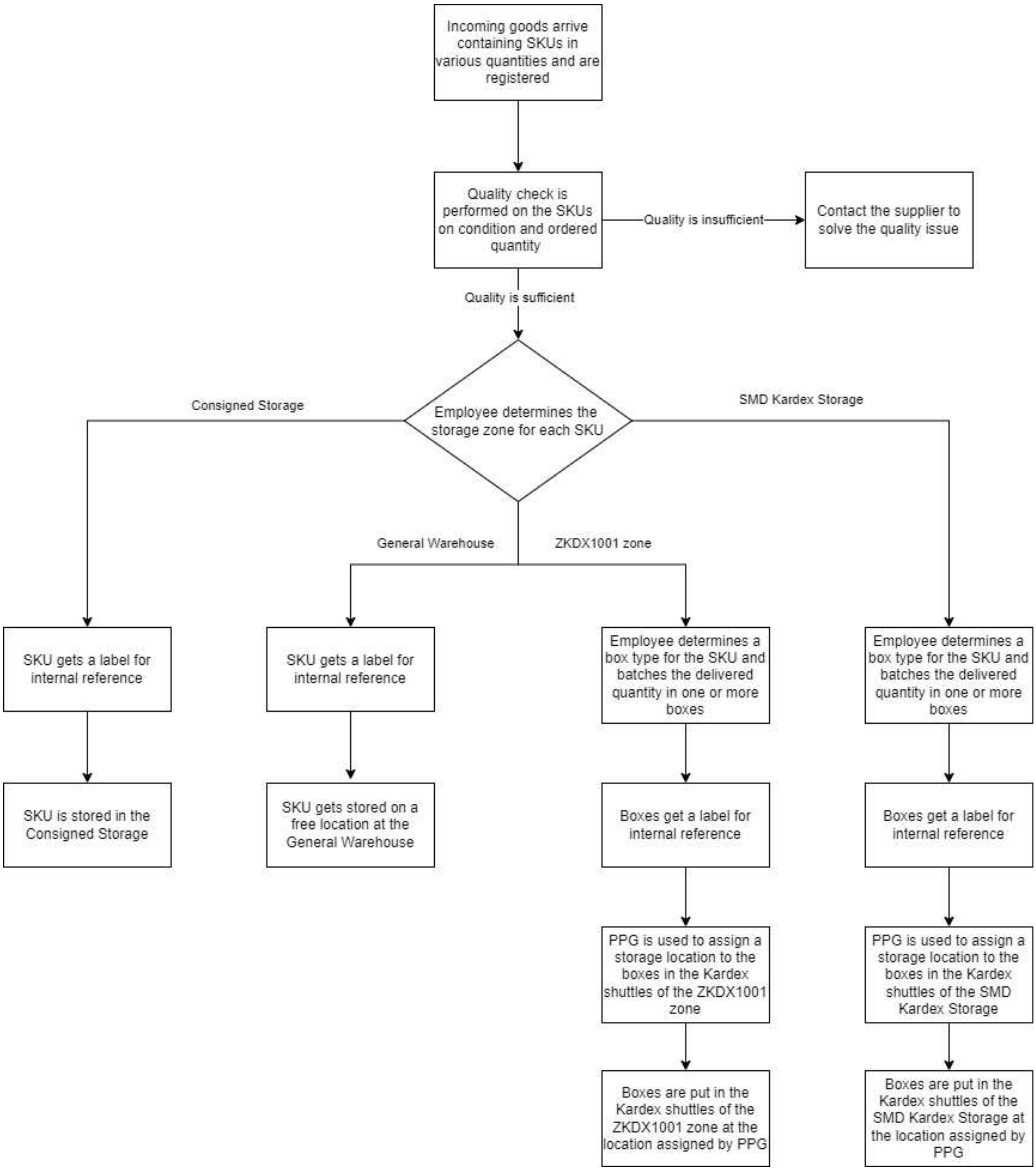

Figure 5: Overview of the reception of incoming goods and storage process

1.3 Problem Statement

This section discusses the problem statement of this research. Section 1.3.1 explains the problem cluster created for this research. Section 1.3.2 discusses the identified core problem.

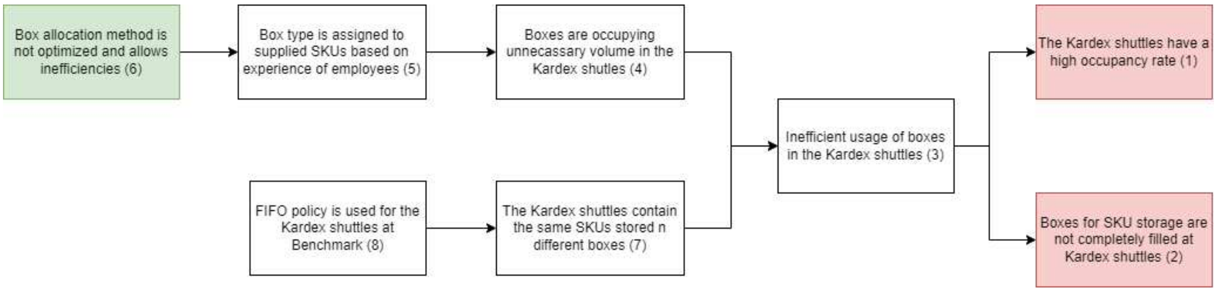

Figure 6: Problem cluster for the Kardex shuttles at Benchmark

1.3.1 Problem Cluster

Figure 6 shows the problem cluster that has been created for the problems that are arising at the Kardex shuttles of Benchmark. Several action problems are faced at the warehouse of Benchmark which are visualized in red. The first action problem faced is the frequent occurrence of a high occupancy rate in the Kardex shuttles (1). The occupancy rate is a metric used at Benchmark to indicate the amount of used boxes to the total amount of boxes available for each box type. A high occupancy rate indicates that currently there are few boxes available in the Kardex shuttles for storage of SKUs in that box type. Secondly, boxes used for SKU storage are not completely filled in the Kardex shuttles (2). In case boxes are not completely filled a part of the box volume could be used for storage of other SKUs. Both action problems are related to inefficient use of boxes in the Kardex shuttles (3).

The inefficient use of boxes in the Kardex shuttles (3) is firstly caused by boxes that are occupying unnecessary volume in the Kardex shuttles (4) due to the box type that is assigned to supplied SKUs based on the experience of employees (5). This way of working allows inefficient box type choices to be made at Benchmark. Therefore, the box allocation method is not optimized and inefficient (6).

Another cause of the inefficient usage of boxes in the Kardex shuttles (3) is that the Kardex shuttles contain the same SKUs stored in different boxes (5). This is caused by the FIFO policy used for the Kardex shuttles at Benchmark (7). The FIFO policy forces the same SKUs to be stored separately and lets the oldest delivery of an SKU leave first due to the expiration dates.

1.3.2 Core Problem

The core problem can be seen as the foundation of all other problems while having no direct cause. Figure 6 provides two potential core problems for Benchmark: the box allocation method is not optimized and allows inefficiencies (8) and the FIFO policy is used within the Kardex shuttles (7). Benchmark has stated that they are currently not considering dropping the FIFO policy for the Kardex shuttles. Therefore, the core problem of this research is surrounding the problem that the box allocation method is not optimized and allows inefficiencies. To indicate the core problem in the problem cluster it is visualized in green as shown in Figure 6. The core problem that needs to be solved during this research has been formulated as follows:

Core problem: The box allocation method used at Benchmark is not optimized and allows employees to make inefficient box type choices.

1.4 Research Design

This section discusses the research design. Section 1.4.1 formulates the research objective. Section 1.4.2 provides the deliverables of this research. Section 1.4.3 defines the scope of this research. Section 1.4.4 explains the methodology used in this research. Section 1.4.5 discusses the research questions and thesis outline

1.4.1 Research Objective

The research objective defines the desired outcome for this research which solves the core problem as defined in Section 1.3.2. The research objective has been formulated as follows:

Research objective: Create a box allocation method that improves storage efficiency in the Kardex shuttles at Benchmark.

In this research, the research objective is to create a box allocation method that improves storage efficiency in the Kardex shuttles. The box allocation method should provide the recommended box type of supplied SKUs and indicate reallocations for boxes stored in inefficient box types in the Kardex shuttles. Reallocations are changes in box types that result in less occupied storage volume for the inventory.

1.4.2 Deliverables

The box allocation method contains two coherent models which are the main deliverables for Benchmark as shown in Figure 7.

1. **Box Division Model**: This model provides Benchmark the ability to choose the recommended box type for supplied SKUs. Furthermore, it can be used to analyze inefficient box type choices for boxes stored in the Kardex shuttles and provide a recommended box type for these boxes. Inefficiencies can be resolved by performing reallocations on the inefficient box type choices. These reallocations change the box types for inefficient box type choices and change the overall box division in the Kardex shuttles. The box division model should provide insights into the change in storage efficiency by performing these reallocations using suitable KPIs.
2. **Reallocation Model**: The reallocation model is aimed at optimizing the operational execution of the reallocations coming from the box division model. It provides insight into the time of completion and total labour costs of executing the changes provided by the box division model.

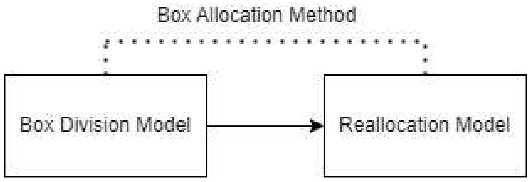

Figure 7: Models used for the Box Allocation Method

1.4.3 Scope

The scope of this research will be improving the storage efficiency in the original ten Kardex shuttles within the ZKDX1001 zone. Therefore, the other Kardex shuttles or storage locations at the warehouse of Benchmark stay out of scope in this research.

1.4.4 Methodology

Figure 8 shows the Design Science Research Methodology (DSRM) introduced by (Peffers et al, 2007). The DSRM is a well-known methodology used for designing solutions where an artifact is being centralized. An artifact is intended to solve identified organizational problems which for Benchmark is the creation box allocation method. Since the creation of a box allocation method involves designing applications the DSRM was determined to be a suitable methodology due to its focus on designing solutions. The steps of the DSRM in the perspective of this research will now be explained.

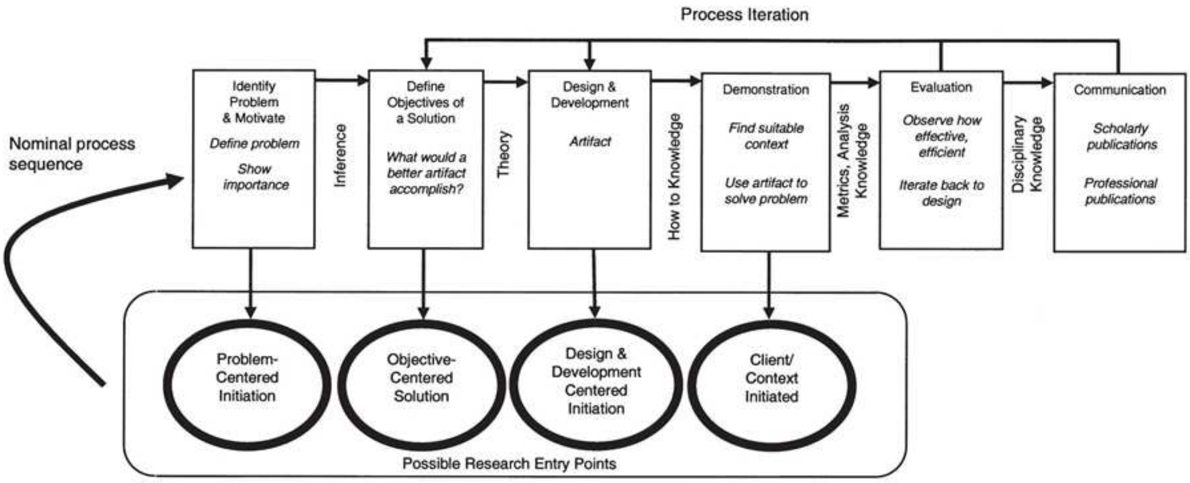

Figure 8: Visualization of the DSRM (Peffers et al., 2007)

Identify Problem & Motivate
The DRSM starts by identifying the problem and motivating why it is a problem. This step is equal to defining the core problem of Benchmark, which is that currently the box allocation method is not optimized and allows employees to make inefficient box type choices as described in Section 1.3.2.

Define the objective of a solution
The objective of a solution refers to the previously defined research objective in Section 1.4.1. The research objective is the creation of a box allocation method that improves storage efficiency in the Kardex shuttles at Benchmark.

Design and development
Design and development centralizes the design of the artifact which becomes the solution of the core problem. The step improves the design of the artifact throughout several iterations after which the artifact can be demonstrated to Benchmark. Chapter 3 investigates the current state of the art methods that can be used in designing the artifact in Chapter 4.

Demonstration
After the design and development of the artifact, its usability is demonstrated. During the demonstration, experiments are performed to show the contribution of the artifact in solving the core problem. These experiments are set up to see how the artifact performs in different circumstances. Chapter 5 demonstrates the artifact using several experiments to observe the impact on the storage efficiency at Benchmark.

Evaluation
The artifact is evaluated using relevant metrics to get clear insight into the performance. The metrics are measurable KPIs that are specifically aimed at the storage efficiency of the Kardex shuttles at Benchmark. Changes can be made to the artifact in case the output is not sufficient yet by comparing the objective to the solution that the artifact created. Chapter 5 contains the evaluation of the artifact.

Communication
Once the artifact has gone through various iterations and provides a solution to the core problem it is communicated to Benchmark. Benchmark will get the outcomes of the solution including a description of the artifact and results expressed in measurable KPIs to see the improvement compared to the current situation. Chapter 6 contains the conclusion of this research and recommends future research.

1.4.5 Research Questions & Thesis Outline

To achieve the research objective a main research question has been formulated which is split up into smaller research questions divided over several chapters.

Main Research Question

Main RQ: " How can more storage efficiency be obtained by using an optimized box allocation method for the Kardex shuttles at Benchmark? "

The main research question will provide a solution to the core problem formulated and will be answered using several sub-questions.

Chapter 2 – Current Situation Analysis

RQ 1: " What does the current inventory layout of the Kardex shuttles look like at Benchmark? "

RQ 2: " Which box types are currently preferred for storage of SKUs in the Kardex shuttles? "

Chapter 2 provides insight into the current situation of the ten Kardex shuttles in the ZKDX1001 zone. These research questions are necessary to get insight into the SKUs stored in the Kardex shuttles and to provide an overview of data that can be used in the development of the box division model and reallocation model.

Chapter 3 – Literature Review

RQ 3: " Which methods could be used by the box division model to improve storage efficiency by defining box types for SKUs? "

RQ 4: " Which algorithms can be used for the creation of the reallocation model? "

Chapter 3 creates an overview of methods that can be used for the box division model and reallocation model. The aim is to come up with suitable methods for designing a box division model and reallocation model at Benchmark.

Chapter 4 – Solution Design

RQ 5: " What would be an efficient box division model for Benchmark? "

RQ 6: " How could the costs of execution reallocations be determined by the reallocation model "

RQ 7: " Which KPIs can be used to measure the change in storage efficiency due to the implementation of the box allocation method for Benchmark? "

Chapter 4 shows the development of the box division model and reallocation model for Benchmark. It aims to create insight into the way the models are created and how Benchmark can use the created solution. The research questions centred on providing insight into the foundation, working, and output of both models.

Chapter 5 – Results and Evaluation

RQ 8: " What would be the impact on the storage efficiency by implementing the box allocation method at Benchmark? "

Chapter 5 evaluates the impact of implementing the box allocation method at Benchmark. This will be done by performing experiments using the models developed box allocation method. After evaluating the results Benchmark a conclusion can be drawn on this research.

Chapter 2 – Current Situation Analysis

This chapter provides the current situation analysis and is related to the following research questions:

RQ 1: " What does the current inventory layout of the Kardex shuttles look like at Benchmark? "

RQ 2: " Which box types are currently preferred for storage of SKUs in the Kardex shuttles? "

Section 2.1 provides insights into the current layout of Kardex shuttles. Section 2.2 contains the SKU analysis of the Kardex shuttles. Section 2.3 provides an inventory analysis of the Kardex shuttles.

2.1 Kardex Shuttles Layout

This section gives insight into the layout of the Kardex shuttles at Benchmark. Section 2.1.1 discusses the six different box types used in the Kardex shuttles. Section 2.1.2 shows the current box division in the Kardex shuttles. Section 2.1.3 contains the carrier setup of the Kardex shuttles. Section 2.1.4 provides insight into the dimensions and usable volume of Kardex shuttles.

2.1.1 Box Types

The Kardex shuttles contain six different types of boxes varying in size as provided in Table 1. As can be seen, the boxes differ from small to large and are suitable for different kinds of SKUs. Furthermore, the available boxes which are not placed in the Kardex shuttles and the net purchasing price of the boxes are shown. Finally, a visual impression of the six different box types is shown in Figure 9.

Table 4: Types of Boxes and their Sizes

Box Type	Dimensions (H x W x D)	Volume	Available Boxes	Net Price
B001	117 x 148 x 200 mm	3.5 L	0	€3.93
B002	170 x 200 x 300 mm	10.2 L	336	€7.00
B003	220 x 300 x 400 mm	26.4 L	320	€9.93
B004	320 x 400 x 600 mm	76.8 L	352	€20.67
B005	120 x 400 x 600 mm	28.8 L	0	€12.07
B006	80 x 89 x 558 mm	4 L	0	€4.73

Figure 9: Overview of the box types starting from B001 (left) to B006 (right)

2.1.2 Overview of Boxes in the Kardex Shuttles

Benchmark has data available on the division of boxes in the Kardex shuttles. Appendix A shows the Location List data set that was used to create an overview of the box division in the Kardex shuttles. Table 5 shows the number of boxes per box type stored in the ten Kardex shuttles. It can be seen that the ten Kardex shuttles contain a total of 13189 boxes.

Table 5: Kardex shuttles box division

Shuttle	B001	B002	B003	B004	B005	B006	Grand Total
SH01	320	335	340	110	70	34	1209
SH02	720	295	280	110	60	34	1499
SH03	560	295	300	110	80	34	1379
SH04	320	374	300	130	50	34	1208
SH05	480	335	240	130	60	34	1279
SH06	480	480	340	100	80	-	1480
SH07	719	320	280	100	70	-	1489
SH08	960	350	180	130	60	68	1748
SH09	240	215	240	170	50	34	949
SH10	240	215	240	170	50	34	949
Grand Total	5039	3214	2740	1260	630	306	13189

2.1.3 Carrier Setup

Carriers are used in the Kardex shuttles to store boxes. Figure 10 shows an example of a carrier retrieved from the Kardex shuttle containing B003 boxes. Appendix B contains specifications on the carriers used in the Kardex shuttles and an impression of an empty carrier without any boxes.

Figure 10: Impression of a retrieved carrier containing B003 boxes

Next to the overview of the box allocation at the Kardex shuttles, it is valuable to see the number of carriers in all Kardex shuttles and the allocated box type. The allocated box type to a carrier is referred to as the carrier type. Table 6 shows an overview of the number of carriers for each carrier type present in all Kardex shuttles. An example from Table 6 would be that there are four carriers in SH01 dedicated to carrier type B001.

Table 6: Carrier setup in the Kardex shuttles

Carrier Type	Amount of carriers in Kardex shuttles									
	SH01	SH02	SH03	SH04	SH05	SH06	SH07	SH08	SH09	SH10
B001	4	9	7	4	6	6	9	11	3	3
B002	8	7	7	9	8	12	8	8	5	5
B003	18	18	16	17	14	20	16	10	12	13
B004	10	7	10	11	11	7	8	12	17	16
B005	7	6	4	4	4	6	5	3	5	5
B006	1	1	0	1	1	0	0	2	1	1

2.1.4 Kardex Shuttles Dimensions and Usable Volume

Figure 11 shows a cross-section of a Kardex shuttle indicating three usable storage zones in blue. These three zones are referred to in this research as the back side, lower front side, and upper front side. These three zones form the total available volume for storage at the Kardex shuttles referred to as the *TotalStorageVolume*. This *TotalStorageVolume* can be split up into *TotalUnoccupiedVolume* and *TotalOccupiedVolume*. Here the *TotalOccupiedVolume* equals all volume occupied by the carriers and boxes which cannot be used anymore. *TotalUnoccupiedVolume r*epresents all volume that is not used and could be used for additional carriers in the Kardex shuttles to increase the number of boxes for specific box types. *TotalUnoccupiedVolume* is a new insight and indicates if the storage capacity can be increased for Benchmark. It can be calculated for each shuttle using the following equation:

$$TotalUnoccupiedVolume = TotalStorageVolume - TotalOccupiedVolume$$

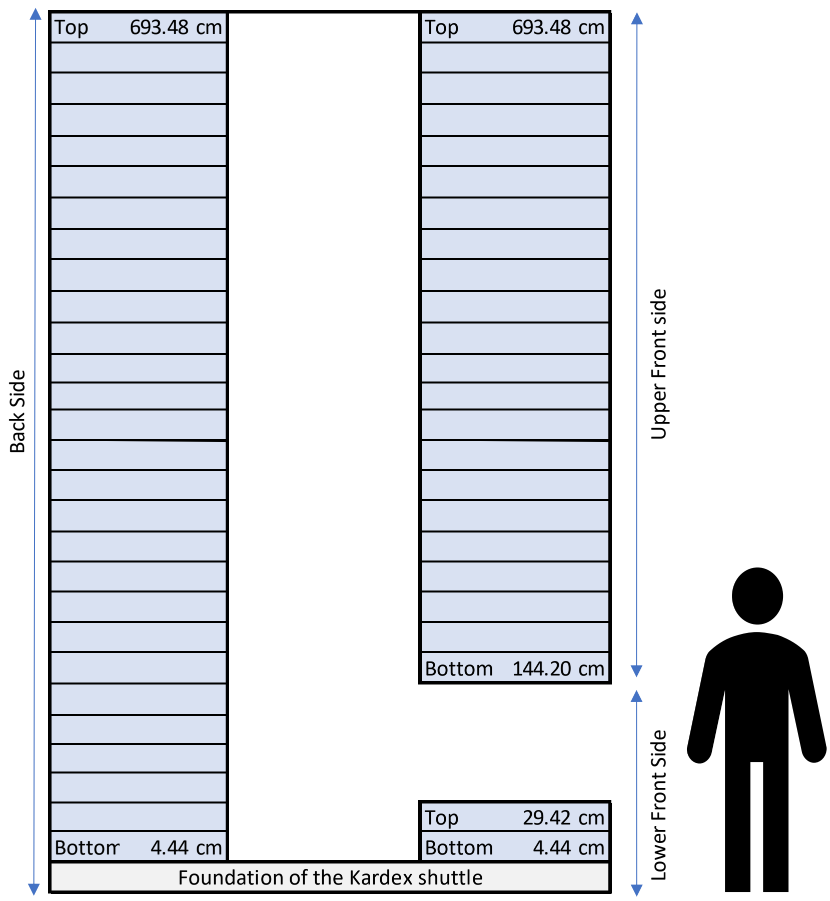

Figure 11: Cross section of a Kardex shuttle showing the three zones

TotalStorageVolume

The first part of the equation for the *TotalUnoccupiedVolume* uses the *TotalStorageVolume* which consist of the volume highlighted in the three blue zones in Figure 11 . The lower front side ranges from 4.44 to 29.42 cm after which there is a gap for the extraction of carriers which is from 27.10 to 141.80 cm. The upper front side starts after the gap at 144.20 cm and contains storage space until the top at 693.48 cm. The back side can be fully utilized for storage, ranging from 4.4 to 693.48 cm. To calculate the *TotalStorageVolume* the following equation has been formulated:

$$TotalStorageVolume$$
$$= StorageVolumeBackSide + StorageVolumeUpperFrontSide$$
$$+ StorageVolumeLowerFrontSide$$

The following equations are used for the calculation of the three storage volumes:

$$StorageVolumeBackSide$$
$$= (BackSideTop - BackSideBottom) * CarrierWidth \ * Carrier\ Depth$$
$$= \frac{(693.48\ cm - 4.44\ cm) * 304.5\ cm * 85cm}{1000000} \approx 17.83\ m^3$$

$$StorageVolumeUpperFrontSide$$
$$= (UpperFrontSideTop - UpperFrontSideBottom) * CarrierWidth * Carrier\ Depth$$
$$= \frac{(693.48\ cm - 144.20\ cm) * 304.5\ cm * 85\ cm}{1000000} \approx 14.22\ m^3$$

$$StorageVolumeLowerFrontSide$$
$$= (BackSideTop - BackSideBottom) * CarrierWidth * Carrier\ Depth$$
$$= \frac{(29.42\ cm - 4.44\ cm) * 304.5\ cm * 85\ cm}{1000000} \approx 0.65\ \ m^3$$

Adding up these three storage volumes gives a *TotalStorageVolume* of approximately 32.70 m^3 per shuttle. In total, the *TotalStorageVolume* of all Kardex shuttles is determined at 326.97 m^3.

TotalOccupiedVolume

The second part of the equation for the *TotalUnoccupiedVolume* uses the *TotalOccupiedVolume*.

Table 7: Volume overview of a single slot in a Kardex shuttle

Slot Volume		
Height per Slot	2.5	cm
Width of Slot	304.5	cm
Depth of Slot	85	cm
Volume Occupied per Slot	64.71	L

Each Kardex shuttle contains 503 slots in the *TotalStorageVolume* that can be used to place carriers in the Kardex shuttles. However, 4 slots cannot be used for placement of carriers due to safety constraints. This means there are 499 usable slots in the *TotalStorageVolume* of the Kardex shuttles. Table 7 shows an overview of the relevant dimensions of a slot which are the height, width, and depth.

Table 8: Overview of used slots for each carrier type

CarrierType	TotalSlots	Volume
B001	6	0.388 m3
B002	8	0.518 m3
B003	10	0.647 m3
B004	14	0.906 m3
B005	6	0.388 m3
B006	8	0.518 m3

Table 8 shows an overview of the slots used by placing each carrier including the boxes for each carrier type in the Kardex shuttle. There can be seen that e.g. placing a B001 carrier occupies 6 slots occupying a volume of 0.388 m3. This is caused by the placement of a carrier itself, the volume of the boxes, and the safety margin that must be considered when placing a carrier. Using this data in combination with the carrier overview as shown in Table 6 allows the calculation of the *TotalOccupiedVolume* per shuttle using the following equation:

$$TotalOccupiedVolume = CarrierOccupiedVolume + UnusableVolume$$

Here the *CarrierOccupiedVolume* is the volume occupied by all carrier type placements in the Kardex shuttles. In each shuttle, the number of carriers per carrier type from Table 6 can be multiplied by the volume of the carrier type as provided in Table 8 and summed to obtain the *CarrierOccupiedVolume*. Furthermore, the *UnsuableVolume* indicates the volume of the four slots that cannot be used in all the Kardex shuttles. Adding up the *CarrierOccupiedVolume* and *UnsuableVolume* for each Kardex shuttle gives *TotalOccupiedVolume* per shuttle as provided in Table 9.

Table 9: Overview of Carrier Occupied Volume and Total Usable Volume

Shuttle	TotalOccupiedVolume	TotalUnoccupiedVolume
SH01	29.89 m3	2.80 m3
SH02	28.21 m3	4.49 m3
SH03	27.56 m3	5.13 m3
SH04	29.51 m3	3.19 m3
SH05	27.82 m3	4.87 m3
SH06	30.41 m3	2.29 m3
SH07	27.44 m3	5.26 m3
SH08	28.21 m3	4.49 m3
SH09	29.64 m3	3.06 m3
SH10	29.38 m3	3.32 m3
Total	288.07 m3	38.90 m3

TotalUnoccupiedVolume

After the *TotalOccupiedVolume* is determined the *TotalUnoccupiedVolume* can be calculated. Table 9 shows the *TotalUnoccupiedVolume* for each Kardex shuttle. Also, the *TotalUnoccupiedVolume* for all Kardex shuttles is determined at 38.90 m3. This value for the *TotalUnoccupiedVolume* shows that currently there is a sufficient amount of volume available that could be used for placing carriers and boxes to increase the storage capacity. Based on Table 8 Benchmark can calculate how many carriers of a carrier type could be added in the Kardex shuttle. An example could be that Benchmark would want to add only B003 carriers. In that case, there can be added 38.90/0.647 = 60 carriers of type B003 in the Kardex shuttles. However, this action would imply a adjustment of the carrier heights in the Kardex shuttles, but shows the potential for adding extra carriers and boxes in the Kardex shuttles.

2.2 Kardex Shuttles SKU Analysis

This section provides insights into the SKUs stored in the boxes in the Kardex shuttles. Section 2.2.1 shows an overview of the SKU analysis. Section 2.2.2 explains the Kardex Inventory Analysis module. Section 2.2.3 shows the Demand & Supply Analysis module. Section 2.2.4 provides insight into the Bin Quantity Analysis module. Section 2.2.5 discusses the use of the Complete Kardex Analysis application.

2.2.1 SKU Analysis Overview

Figure 12 shows an overview of the Complete Kardex Analysis application which is a data analysis program developed for this research. The purpose of this analysis is to provide detailed information on the SKUs stored in the Kardex shuttles that can be used in the development of the box allocation method. The program is created using Python and runs several modules when being executed. In the Complete Kardex Analysis, three modules are used: Kardex Inventory Analysis, Demand & Supply Analysis, and Bin Quantity Analysis.

- **Kardex Inventory Analysis**: The Kardex Inventory Analysis module provides insights into all SKUs stored in the Kardex shuttles and is discussed in Section 2.2.2.
- **Demand & Supply Analysis**: The Demand & Supply Analysis module creates an impression of the SKU demand and supply patterns and is explained in Section 2.2.3.
- **Bin Quantity Analysis**: The Bin Quantity Analysis module calculates the number of items that fit in each box type for all SKUs and is provided in Section 2.2.4.

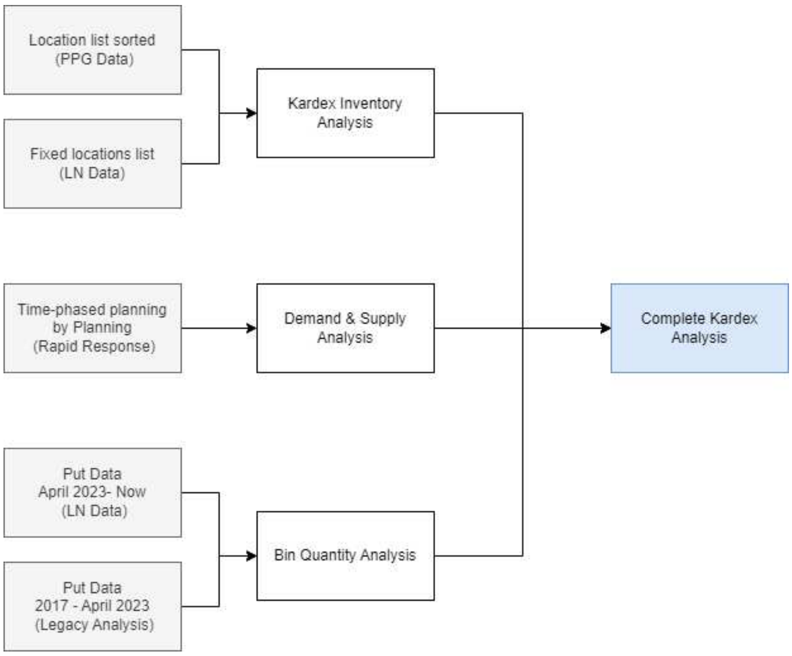

Figure 12: Overview of Complete Kardex Analysis

All three modules make use of input data that has been collected from LN, Rapid Response, and PPG as shown in Figure 12. The Kardex Inventory Analysis module utilizes the Location List and Fixed Location List. The Location list was also used for the overview of boxes as shown in Table 5 and is provided in Appendix A. The Location List contains information on all currently stored SKUs at Benchmark. Appendix C provides an overview of the Fixed Location List, containing all SKUs that have their fixed storage location in the Kardex shuttles. The Demand & Supply Analysis module utilizes the Time-Phased Planning report as shown in Appendix D, which is crucial for the analysis of demand and supply of SKUs. Lastly, the Bin Quantity Analysis module uses Put Data extracted from PPG as shown in Appendix E. The Put Data contains insights into which box types and quantities SKUs are being put into the Kardex shuttles when supplied to Benchmark.

2.2.2 Kardex Inventory Analysis Module

The Kardex Inventory Analysis is a module used by the Complete Kardex Analysis which should provide insight into all SKUs stored in the Kardex shuttles. As mentioned earlier, the module utilizes the Location List in Appendix A and the Fixed Location List in Appendix C to obtain a complete overview of the SKUs stored in the Kardex shuttles. The Kardex Inventory Analysis module applies some data cleaning to only keep the relevant data from the Location List. This is done because the original Location List contained data on SKUs stored outside of the ten Kardex shuttles at the ZKDX1001 zone. Figure 13 shows the output file “Export Kardex SKUs” which contains all relevant data on the SKUs stored in the Kardex shuttles at the ZKDX1001 zone from the Location List.

1	SKU	Current quantity	Storage unit	Carrier	Put date	Bin	Location	Handling Unit
2	ASM4022_438_75306-LF	128	SH06	8	03/31/2023	B002	SH06-008-003-03	
3	ASM4022_711_85891-LF	8	SH04	6	12/15/2023	B002	SH04-006-009-01	B002-0002
4	ASM4022_438_09089-LF	1000	SH02	1	02/05/2024	B001	SH02-001-020-01	B001-3153
5	ASM4022_685_40902-LF	10	SH06	18	02/20/2024	B003	SH06-018-003-01	B003-1361
6	ASM4022_693_80701-LF	3	SH02	46	03/31/2023	B001	SH02-046-013-02	
7	ASM4022_668_81621-LF	30	SH05	9	03/21/2024	B002	SH05-009-005-03	B002-1181
8	ASM4022_438_34209-LF	73	SH03	8	03/04/2024	B002	SH03-008-009-04	B002-2057
9	ASM4022_472_56922-LF	15	SH08	38	03/05/2024	B004	SH08-038-005-01	B004-0663
10	ASM4022_711_21431-LF	8	SH07	30	04/08/2024	B003	SH07-030-004-01	B003-3132
11	ASM4022_685_71132-LF	15	SH04	21	04/23/2024	B003	SH04-021-002-01	B003-1911
12	ASM4022_476_23022-LF	8	SH05	40	02/27/2024	B003	SH05-040-010-02	B003-1489
13	ASM4022_476_01401-LF	6	SH08	14	03/22/2024	B003	SH08-014-010-01	B003-2600
14	ASM4022_660_28752-LF	1	SH01	9	03/31/2023	B002	SH01-009-08-01	
15	ASM4022_698_29811-LF	8	SH04	7	04/10/2024	B002	SH04-007-003-01	B002-3023
16	ASM4022_472_65202-LF	10	SH02	4	06/22/2023	B002	SH02-004-010-03	B002-3728
17	ASM4022_438_41288-LF	1	SH07	11	03/31/2023	B001	SH07-011-003-03	

Figure 13: Screenshot of the “Export Kardex SKUs” of the Kardex Inventory Analysis

Also, there is created a summary file called “Export SKU Summary” that contains an overview of the total quantity in stock for all SKUs as can be seen in Figure 14. This is done by collecting all SKUs stored in the Kardex shuttles from the Fixed Locations List and calculating the total current quantity of these SKUs based on the Location List. This overview is useful since the SKU could be stored at several different locations and it shows the total quantity available instead of the individual quantity of an SKU at a certain location as in the “Export Kardex SKUs “.

1	SKU	Current quantity
2	ASM0051_102_00002-LF	48
3	ASM0051_103_00111-LF	2
4	ASM0051_103_00121-LF	7
5	ASM0051_103_00131-LF	1
6	ASM0051_103_00141-LF	20
7	ASM0051_103_00151-LF	2
8	ASM0051_103_00161-LF	1
9	ASM0051_103_00171-LF	1
10	ASM0051_103_00181-LF	1
11	ASM0051_103_00191-LF	1
12	ASM0051_103_00201-LF	2
13	ASM0051_103_00231-LF	2
14	ASM0051_103_00241-LF	1
15	ASM0051_103_00251-LF	1
16	ASM0051_103_00271-LF	1
17	ASM0051_103_00301-LF	2
18	ASM0051_103_00321-LF	3
19	ASM0051_103_00331-LF	3
20	ASM0051_103_00341-LF	1

Figure 14: Screenshot of the "Export SKU Summary" of the Kardex Inventory Analysis

Lastly, the Kardex Inventory Analysis module exports the "Complete Kardex List" which contains an overview of all unique SKUs that are stored in the Kardex shuttles present in the Location List and Fixed Location List as can be seen in Figure 15.

1	SKU
2	ASM0051_102_00002-LF
3	ASM0051_103_00111-LF
4	ASM0051_103_00121-LF
5	ASM0051_103_00131-LF
6	ASM0051_103_00141-LF
7	ASM0051_103_00151-LF
8	ASM0051_103_00161-LF
9	ASM0051_103_00171-LF
10	ASM0051_103_00181-LF
11	ASM0051_103_00191-LF
12	ASM0051_103_00201-LF
13	ASM0051_103_00231-LF
14	ASM0051_103_00241-LF
15	ASM0051_103_00251-LF

Figure 15: Screenshot of the "Complete Kardex List" of the Kardex Inventory Analysis

2.2.3 Demand & Supply Analysis Module

The Demand & Supply Analysis module is used to gain insight into the demand and supply patterns of SKUs. As discussed, the module makes use of the Time-Phased planning report as shown in Appendix D. The main function of this module is to filter the data of the Time-Phased planning report by removing irrelevant data of SKUs that are stored outside of the ten Kardex shuttles at the ZKDX1001 zone. This is done by only collecting data on SKUs placed in the "Complete SKU List" file from the Kardex Inventory Analysis as introduced in Section 2.2.2. Furthermore, the modules adjust the timespan of the Time-Phased planning report to one month such that there can only be demand and supply for the next month. Benchmark addresses that the absolute demand and supply numbers for the next month of SKUs should be leading for the developed models. This is because the models will run monthly and actions should be taken upfront based on knowledge at the beginning of a month. Figure 16 shows an impression of the "DemandAnalysis" output from the module where the total demand of all SKUs in the Kardex shuttles has been listed for the next month.

1	SKU	TotalDemand
52	ASM0051_240_01981-LF	3
53	ASM0051_240_01981-P	24
54	ASM0051_240_02281-LF	14
55	ASM0051_240_02292-LF	
56	ASM0051_240_02321-LF	
57	ASM0051_240_02521-P	1
58	ASM0051_240_02542-LF	
59	ASM0051_240_02542-P	1
60	ASM0051_240_02642-V91	
61	ASM0051_240_02643-P	4
62	ASM0051_240_02661-P	2
63	ASM0051_240_02791-P	
64	ASM0051_240_02913-P	10

Figure 16: Screenshot of the "DemandAnalysis" from the Demand & Supply Analysis

Next to the total demand overview, an overview containing all demand for SKUs per day in the month has been made. This is exported to the “FullDemand” file by the module as can be seen in Figure 17.

1	SKU	KardexArticle	Demand	Due Date	Full Date
2	ASM0051_240_00571-P	1	4	06/19/2024 0:00	06/19/2024 0:00
3	ASM0051_240_00571-P	1	4	06/19/2024 0:00	06/19/2024 0:00
4	ASM0051_240_00571-P	1	4	06/24/2024 0:00	06/24/2024 0:00
5	ASM0051_240_00571-P	1	4	06/24/2024 0:00	06/24/2024 0:00
6	ASM0051_240_00691-LF	1	2	06/03/2024 0:00	06/03/2024 0:00
7	ASM0051_240_00691-LF	1	2	06/03/2024 0:00	06/03/2024 0:00
8	ASM0051_240_00691-LF	1	2	06/06/2024 0:00	06/06/2024 0:00
9	ASM0051_240_00691-LF	1	2	06/10/2024 0:00	06/10/2024 0:00
10	ASM0051_240_00691-LF	1	2	06/12/2024 0:00	06/12/2024 0:00
11	ASM0051_240_00691-LF	1	2	06/12/2024 0:00	06/12/2024 0:00
12	ASM0051_240_00691-LF	1	2	06/13/2024 0:00	06/13/2024 0:00
13	ASM0051_240_00691-LF	1	2	06/18/2024 0:00	06/18/2024 0:00
14	ASM0051_240_00691-LF	1	2	06/20/2024 0:00	06/20/2024 0:00
15	ASM0051_240_00691-LF	1	2	06/21/2024 0:00	06/21/2024 0:00
16	ASM0051_240_00691-LF	1	2	06/25/2024 0:00	06/25/2024 0:00
17	ASM0051_240_00691-LF	1	2	06/26/2024 0:00	06/26/2024 0:00
18	ASM0051_240_00691-LF	1	2	06/28/2024 0:00	06/28/2024 0:00

Figure 17: Screenshot of the "Full Demand" from the Demand & Supply Analysis

Next to the demand overviews, the module also provides an overview of all supply for SKUs per day in the month. Figure 18 shows the “SupplyAnalysis” file which contains all supplies for the SKUs stored at the Kardex shuttles. Note that it could be the case that several supplies of similar SKUs are scheduled in the upcoming month. Therefore, a rank has been made next to the delivery date indicating the order of deliveries starting from 1 for the earliest supply. Finally, in case the supply field is blank the SKU won’t be supplied in the upcoming month.

1	SKU	Supply	Due Date	Rank	Full Date
314	ASM4022_438_20759-LF	19	6	1	06/19/2024 0:00
315	ASM4022_438_20961-LF				
316	ASM4022_438_20972-LF				
317	ASM4022_438_25122-LF	51	6	1	06/03/2024 0:00
318	ASM4022_438_25122-LF	37	6	2	06/05/2024 0:00
319	ASM4022_438_25122-LF	51	6	3	06/10/2024 0:00
320	ASM4022_438_25122-LF	85	6	4	06/18/2024 0:00
321	ASM4022_438_25122-LF	5	6	4	06/18/2024 0:00
322	ASM4022_438_25122-LF	28	6	5	06/24/2024 0:00
323	ASM4022_438_25122-LF	30	6	5	06/24/2024 0:00
324	ASM4022_438_25122-LF	28	6	6	06/27/2024 0:00
325	ASM4022_438_25188-LF				
326	ASM4022_438_30096-LF				
327	ASM4022_438_30098-LF				

Figure 18: Screenshot of the "SupplyAnalysis" from the Demand & Supply Analysis

2.2.4 Box Quantity Analysis Module

The Box Quantity Analysis module provided insights into the maximum quantity of an SKU that fits in each box type. As discussed the input data for this module is the Put Data coming from PPG as shown in Appendix E. Put data can be used to observe the maximum quantity of SKUs that fit into each box type. For all SKUs, historic puts and the maximum quantity of the puts in each box type is calculated. This provides the maximum quantity that fits in each box type for all SKUs.

1	SKU	B001	B002	B003	B004	B005	B006
50	ASM0051_240_01941-P	0	0	4	11	4	0
51	ASM0051_240_01971-LF	0	0	1	2	0	0
52	ASM0051_240_01981-LF	0	0	5	19	5	0
53	ASM0051_240_02281-LF	0	0	4	6	6	0
54	ASM0051_240_02292-LF	0	0	1	0	1	0
55	ASM0051_240_02321-LF	0	0	1	1	1	0
56	ASM0051_240_02521-P	0	0	1	5	0	0
57	ASM0051_240_02542-P	0	0	0	15	0	0
58	ASM0051_240_02642-V91	0	0	0	0	7	0
59	ASM0051_240_02661-P	0	0	6	1	0	0
60	ASM0051_240_02791-P	0	0	4	6	0	0
61	ASM0051_240_02913-P	0	0	11	0	10	0
62	ASM0051_240_02971-P	0	0	2	0	2	0
63	ASM0051_240_03001-P	0	0	5	0	0	0
64	ASM0051_240_03042	0	0	5	15	0	0

Figure 19: Screenshot of "SKU Bin Quantity" from the Box Quantity Analysis

Figure 19 shows an impression of the "SKU Bin Quantity" output file which contains the maximum quantity of all SKUs that fit in each box type. Note that due to a change in PPG software the current Put Data only goes back until April 2023. To get a more complete picture of the Put Data a merge has been performed on older Put Data resulting in the output "Merged Bin Quantities" which is based on Put Data coming from the older PPG software before April 2023 and the current PPG software.

2.2.5 Complete Kardex Analysis

The Complete Kardex Analysis application is created to execute all modules and merge data from these modules. After merging and formatting data coming from these three modules an output file is created called "KardexInventoryAnalysis". This file contains all required information for the SKUs stored in the Kardex shuttles as visualized in Figure 20. A valuable addition is the "Rank" indicating the earliest delivery date, where 1 is the earliest delivery.

1	SKU	Current quantity	Storage unit	Carrier	Put date	Bin	Location	Handling Unit	MOQ	EOQ	TotalSKUInventory	TotalDemand	QB001	QB002	QB003	QB004	QB005	QB006	Rank	TotalSKUs
2	ASM4022_438_75306-LF	128	SH06	8	03/31/2023	B002	SH06-008-003-03	0	1	1	128	4	50	0	0	0	0	0	1	1
3	ASM4022_711_85891-LF	8	SH04	6	12/15/2023	B002	SH04-006-009-01	B002-0002	1	1	8	2	1	10	0	0	0	0	1	1
4	ASM4022_438_09089-LF	1000	SH02	1	02/05/2024	B001	SH02-001-020-01	B001-3153	100	1	1021	131	1000	0	0	0	0	0	2	2
5	ASM4022_685_40902-LF	10	SH06	18	02/20/2024	B003	SH06-018-003-01	B003-1361	1	1	15	10	0	0	15	0	0	0	2	2
6	ASM4022_693_80701-LF	3	SH02	46	03/31/2023	B001	SH02-046-013-02	0	10	1	3	0	10	0	0	0	0	0	1	1
7	ASM4022_668_81621-LF	30	SH05	9	03/21/2024	B002	SH05-009-005-03	B002-1181	30	1	32	6	0	30	33	0	0	0	2	2
8	ASM4022_438_34209-LF	73	SH03	8	03/04/2024	B002	SH03-008-009-04	B002-2057	1	24	977	507	61	116	0	288	288	0	4	11
9	ASM4022_472_56922-LF	15	SH08	38	03/05/2024	B004	SH08-038-005-01	B004-0663	15	1	31	0	0	0	15	0	0	0	2	3
10	ASM4022_711_21431-LF	8	SH07	30	04/08/2024	B003	SH07-030-004-01	B003-3132	1	1	8	5	0	0	10	0	0	0	1	1
11	ASM4022_685_71132-LF	15	SH04	21	04/23/2024	B003	SH04-021-002-01	B003-1911	1	1	44	21	0	2	15	30	0	0	4	4
12	ASM4022_476_23022-LF	8	SH05	40	02/27/2024	B003	SH05-040-010-02	B003-1489	5	1	23	10	5	6	10	0	0	0	3	3
13	ASM4022_476_01401-LF	6	SH08	14	03/22/2024	B003	SH08-014-010-01	B003-2600	1	1	133	121	0	2	13	5	5	0	11	24
14	ASM4022_660_28752-LF	1	SH01	9	03/31/2023	B002	SH01-009-08-01	0	1	1	1	0	0	2	5	0	0	0	1	1
15	ASM4022_698_29811-LF	8	SH04	7	04/10/2024	B002	SH04-007-003-01	B002-3023	1	1	79	48	0	8	33	70	0	0	4	7
16	ASM4022_472_65202-LF	10	SH02	4	06/22/2023	B002	SH02-004-010-03	B002-3728	45	1	100	0	4	42	120	0	0	0	1	3
17	ASM4022_438_41288-LF	1	SH07	11	03/31/2023	B001	SH07-011-003-03	0	1	1	72	0	10	50	0	0	0	0	4	9
18	ASM4022_438_40734-LF	11	SH06	5	02/15/2024	B002	SH06-005-007-03	B002-0162	10	1	15	4	0	20	0	0	0	0	2	2
19	ASM4022_438_33382-LF	53	SH08	7	03/31/2023	B001	SH08-007-009-02	0	1	1	343	1	72	0	0	0	0	0	7	8
20	ASM4022_478_00740-LF	4	SH07	2	04/15/2024	B001	SH07-002-003-01	B001-1028	1	1	14	18	10	5	0	0	0	0	2	4
21	ASM4022_693_97832-LF	4	SH09	30	04/03/2024	B006	SH09-030-034-01	B006-0141	1	1	24	16	0	0	8	0	0	10	2	4

Figure 20: Screenshot of "KardexInventoryAnalysis" from the Complete Kardex Analysis

2.3 Box Analysis of the Kardex Shuttles

This section contains insights into the boxes stored at the original ten Kardex shuttles in the ZKDX1001 zone. Section 2.3.1 discusses the occupancy rate used for boxes in the Kardex shuttles. Section 2.3.2 explains the issue of box capacity usage.

2.3.1 Occupancy Rate

An important insight into the Kardex shuttles is the occupancy rate. The occupancy rate shows the number of occupied boxes in comparison to the total boxes available for each box type in the Kardex shuttles. Therefore, the following equation calculation of the occupancy rate per box type:

$$OccupancyRatePerBoxType = \frac{OccupiedBoxesPerBoxType}{TotalAvailableBoxesPerBoxType}$$

Research by Horselenberg (2023) created insight into the current occupancy rate of the Kardex shuttles at Benchmark by use of a tool. Before the acquisition of the two additional Kardex shuttles, the occupancy rate was on average 98% at the original ten Kardex shuttles. This occupancy rate is undesired and should be decreased. The new Kardex shuttles are expected to decrease the pressure on the original ten Kardex shuttles since boxes will be placed from the original to the new Kardex shuttles. However, Benchmark wants to prevent the Kardex shuttles from getting a high occupancy rate again.

2.3.2 Box Capacity Usage

An issue that Benchmark is facing in the boxes stored on the Kardex shuttles is the low box capacity usage. The box capacity usage represents the percentage that boxes are filled. In case there is a low box capacity usage for a box this means that the box contains fewer items than would be possible. Appendix F shows an overview of the box capacity usage by the SKUs placed on the original ten Kardex shuttles. For each SKU there is divided the current quantity of each SKU coming from the "KardexInventoryAnalysis" as shown in Section 2.2.5, by the maximum quantity of the SKU in that fits in the box type from the "SKU Bin Quantity" file as provided in Section 2.2.4. This results in the box capacity usage for each box placed in the Kardex shuttles. This allows the calculation of the average box capacity usage of all boxes in the Kardex shuttles, which is determined at 64.8%. This percentage shows that more than one third of the volume in the boxes is not used which is undesired.

2.4 Conclusion

This chapter showed insights into the current situation surrounding the SKUs stored in the Kardex shuttles. A clear overview of boxes and carriers placed in the Kardex shuttles was provided. Furthermore, the dimensions and usable volume of the Kardex shuttles were calculated. Also, a data analysis is performed on the SKUs stored in the Kardex shuttles providing a complete picture of the data surrounding these SKUs. The output files created by the data analysis can be used in the development of both models. Lastly, the occupancy rate and box capacity usage were introduced as the metrics for the boxes stored in the Kardex shuttles.

Chapter 3 – Literature Review

This chapter discusses the literature review performed to get insight into scientific knowledge that applies to this research and is related to the following research questions.

RQ 3: " Which methods could be used by the box division model to improve storage efficiency by defining box types for SKUs? "

RQ 4: " Which algorithms can be used for the creation of the reallocation model? "

Section 3.1 discusses several storage allocation methods. Section 3.2 dives into the available literature on multiple VLM storage allocation methods. Section 3.3 concludes which storage method will be suitable for Benchmark. Section 3.4 discusses methods to decrease occupied volume by box type choices. Section 3.5 explains information on heuristics needed for creating a model for reallocations.

3.1 Storage Allocation Methods

As discussed in Section 1.2.4, Benchmark uses a random storage allocation method driven by the PPG software for the Kardex shuttles. This decision was made based on outcomes from research by Jansman (2014). However, at that time only six Kardex shuttles were present and no storage capacity issue was faced. Since implanting the box allocation method results in changes for the box division in the Kardex shuttles, Benchmark finds it valuable to investigate a different storage allocation method that could improve order-picking time. In the literature, a wide variety of storage allocation methods are available. This section discusses several storage allocation methods that might be suitable for Benchmark.

Random Storage

Random storage is one of the less complex methods of storing items in warehouses. This storage allocation method aims at providing maximum flexibility in locations where goods can be stored. The random storage allocation method will allow boxes to be placed anywhere in the storage racks of a warehouse with an equal probability (Çelk & Süral, 2013). Random-based storage has the advantage of being more resilient to demand changes than other storage allocation methods. As discussed, earlier research by Jansman (2014) resulted in an initial implementation of a random storage assignment for the Kardex shuttles when they were initially installed.

Closest Open Location Storage

A storage allocation method having a similar approach to the random storage policy is the closest open location storage method (COL). The COL makes use of input/output point selection during the storage of SKUs based on the current location where an operator is standing in the warehouse (Park & Lee, 2007). This means that items are placed at the nearest open location in the storage racks when storing incoming parts in boxes to the storage racks. COL is according to the literature not one of the most suitable storage allocation methods in many cases. Therefore, it will not likely lead to an implementation of this method at the warehouse of Benchmark.

Dedicated Storage

Another widely adopted storage allocation method is the dedicated storage method. Dedicated storage allocation creates fixed locations for SKUs to be stored in warehouses. An often referred to advantage of this method is data-handling efficiency which is caused by the location of SKUs won't change in comparison to other storage allocation methods (Lee & Elsayed, 2005). However, the use of dedicated storage does not seem to be the most efficient when it comes to reducing used capacity. Since fixed locations are used for all SKUs, dedicated storage will not consider saving space if SKUs are currently not in stock. In the occurrence of such a situation, there will still be reserved space for a SKU.

Turnover-Based Storage
Turnover-based storage allocation stores SKUs based on individual turnover (Yu & De Koster, 2013). This storage allocation method makes sure to allocate items with higher turnovers to the input/output point. Turnover-based storage will perform a classification on the turnover and place SKUs with a higher turnover value closer to the input/output point. The method makes use of an ABC classification to determine the turnover value. In this classification method, A items are the most valuable items, B items are medium items, and C items represent the lowest turnover items. The ABC classification sometimes refers to D items which represent items that don't sell at all. Furthermore, A items represent 20% of the total inventory while contributing to 80% of the total turnover. B items are 20% to 30% of the inventory, contributing to 10% of the turnover. Furthermore, C items are all other items that contribute to the turnover value with a total contribution of 10% of the total turnover. In case the ABC classification considers D items, these do not contribute to the total turnover.

Class-Based Storage
Class-based storage is a storage allocation method that stores SKUs based on picking activity (Petersen et al., 2004). The main point of interest in class-based storage is the picking frequency of SKUs. The warehouse will be divided into different sections where an SKU will be stored randomly in one of the dedicated sections. In these sections, the items with higher picking frequency are placed closer to the input and output points. A commonly used classification method is the ABC classification as discussed in turnover-based storage. The difference is that A relates to the most commonly picked items, B to the medium-picked items, and C to the least picked items. Again, there could be talked about the D classification which relates to items that are not picked anymore. Class-based storage seems to be extremely useful for an increase in order picking speed since it allocates SKUs based on the picking rate.

3.2 Storage Allocation Methods for a Multiple VLM Setup

Although there is a wide range of literature available on storage allocation methods in general there seems to be a lack of academic research on storage allocation for a setup of multiple VLMs which is relevant for Benchmark. However, literature is available on implementing different storage allocation methods in a single VLM.

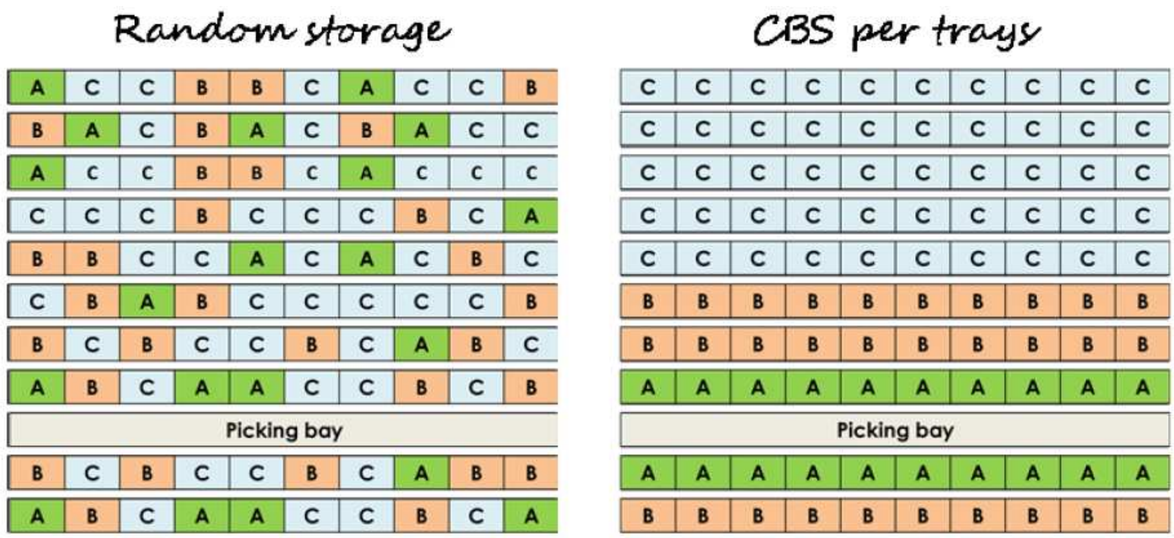

Figure 21: Illustration of different storage allocation methods in a VLM (Battini et al., 2016)

Research by Battini et al. (2016) compares the random storage allocation method in comparison to the class-based storage for usage in a single VLM. Figure 21 visualizes these two different storage allocation methods for a single VLM. The research showed that implementing a class-based storage allocation method could lead to a significant increase in order-picking speed. If in the order-picking process batching is combined with class-based storage significant improvements are found in the order-picking speed. A limitation of this research is that it is focused on a single VLM system. However, the findings from this research could be transferred to an extended method for a setup containing Multiple VLMs which is the case at Benchmark.

3.3 Determining the Storage Allocation Method

Based on the provided storage allocations the class-based storage method is determined to be the most suitable storage allocation. This is due to the focus on efficient order-picking times while maintaining flexibility for random storage in the different storage zones as shown in Figure 21. This conclusion was also formed in earlier research performed at Benchmark for potential better storage allocation methods (Tenhagen, 2018). To verify this conclusion, Kardex Remstar was contacted during this research to provide feedback on using class-based storage in the Kardex shuttles. Kardex Remstar concluded that class-based storage would only be profitable if Kardex shuttles were higher than 10 meters. As explained in Section 2.1.4 the Kardex shuttles of Benchmark are approximately 7 meters high, which makes class-based storage not a profitable option for Benchmark. Therefore, Kardex Remstar advises sticking to the random storage allocation method instead of changing to a different storage allocation method. This conclusion is taken into account for this research and therefore no further investigation into a different storage allocation method will be performed.

3.4 Decrease Occupied Volume in a VLM by a Box Allocation Method

After performing a broad literature review there seems to be no study available focusing on decreasing the occupied volume by performing box-type changes for VLMs. Most research aims at improving the throughput of boxes by increasing order-picking speed. Earlier research at Benchmark increased the efficiency of the order-picking sequences by using optimization techniques (Horselenberg, 2023). During this research, the throughput model created by Lenoble et al. (2018) was used for calculations of the order picking time. This model could be valuable in the creation of the reallocation model which optimizes the completion time of reallocations across multiple VLMs. Furthermore, the model of Đukić et al. (2015) provided the basis to calculate the throughput in a VLM as provided in the research of Lenoble et al. (2018). Since there is a lack of research in the academic world on decreasing the occupied volume in VLMs by implementing a box allocation method this research aims to fill this research gap.

3.5 Determining the Order of Reallocations

This section dives into methods that could be used for the reallocations model to schedule the reallocations coming from the box division model. Section 3.5.1 explains scheduling reallocations as a combinatorial optimization problem. Section 3.5.2 discusses the use of constructive heuristics. Section 3.5.3 explains the use of improvement heuristics.

3.5.1 Combinatorial Optimization Problem

The order of executing reallocations coming from the box division model is a combinatorial optimization problem. Combinatorial optimization problems are NP-hard problems which means that they cannot be solved in polynomial time (Toth, 2000) and require heuristics to obtain a solution. The reallocation model can be seen as a job shop scheduling problem that aims to minimize the completion time of all order pickers. Therefore, a scheduling model can be created for the reallocations and a solution can be found using heuristics. Ardjmand et al. (2018) introduced a scheduling model that can be used as a foundation for the development of the reallocation model.

3.5.2 Constructive Heuristics

Literature offers a wide range of constructive heuristics that could be used for the reallocation model. Constructive heuristics are aimed at creating an initial feasible solution using an algorithm. There are many options for constructive heuristics as long as the solution remains feasible, meaning it can be created based on given constraints. The construction heuristic for reallocations should create a feasible solution based on the output of the box division model. A couple of constructive heuristics can be considered for the reallocation model.

Nearest Neighbour

One of the most common constructive heuristics for combinatorial organization problems is the nearest neighbour heuristic. This heuristic operates by iteratively selecting the closest unvisited location, thereby constructing a feasible solution (Rahman & Parvez, 2021). In the context of reallocating boxes within a warehouse, the objective is to find the most efficient way to move boxes to new locations while adhering to constraints such as capacity and compatibility. Utilizing the nearest neighbour heuristic, one would start by picking an initial box that requires reallocation. The next step is to identify the nearest feasible box that can be reallocated. Note that the nearest box can be defined based on various criteria, such as distance or time.

The process is then repeated for the next box, always selecting the nearest feasible location that has not yet been utilized. This iterative process continues until all boxes have been reallocated. While the nearest neighbour heuristic is valued for its simplicity and efficiency in generating initial solutions, it is important to note that it may not always produce the most optimal solution, particularly for larger and more complex problems. Nonetheless, it provides a practical starting point for further optimization efforts.

Farthest Neighbour

The farthest neighbour heuristic is another method applied to box picking within a warehouse to ensure a balanced and even distribution of picking tasks. This heuristic involves selecting the farthest unvisited location for each subsequent pick. In implementing this approach, one begins by picking a box at an initial location. Instead of proceeding to the nearest box for the next pick, the heuristic identifies the farthest feasible box within the warehouse, based on criteria such as distance and accessibility (Agarwal et al., 1992). The picker then moves to this farthest location for the subsequent pick. By consistently selecting the farthest location, the farthest neighbour heuristic effectively distributes picking tasks across the entire warehouse. This approach prevents congestion and promotes an even workload distribution, reducing the likelihood of bottlenecks in certain areas. While the farthest neighbour heuristic may not always result in the shortest total picking route, it provides a robust initial strategy for balancing picking tasks. This method serves as a practical foundation that can be further refined through optimization techniques to enhance overall efficiency and effectiveness.

Randomized Allocation

Another constructive heuristic is the randomized allocation heuristic. This method introduces a level of randomness to the box-picking or allocation process, providing a diverse range of feasible solutions that can be refined later (Gil-Borrás et al., 2021). In this approach, each box to be allocated or picked is assigned a location randomly from a pool of feasible options that meet the necessary constraints such as capacity, compatibility, and accessibility.

Instead of systematically choosing the nearest or farthest location, the heuristic randomly selects a random feasible location for this box. The criteria for feasibility are still adhered to, ensuring that the location can accommodate the box in terms of space and other relevant factors. After placing the box in a randomly selected location, the process is repeated for all other boxes. By incorporating randomness, this heuristic can help avoid local optima and explore a wider solution space, potentially leading to more innovative and effective configurations. While randomized allocation may not always provide the most efficient initial solution, it serves as a valuable method for generating diverse starting points.

3.5.3 Improvement Heuristics

Improvement heuristics are used to find a better solution based on the constructive heuristics. For the solution, the improvement heuristic will use operators to change item allocation in the Kardex shuttles. A solution could be accepted if it improved the initial solution from the constructive heuristic. In this case, improvement can refer to decreasing the completion time of reallocations. If the completion time of the reallocations is lower than the initial solution it can be accepted and will be kept as the best solution yet.

Local Search

Local search is an improvement heuristic that focuses on exploring the neighbourhood of the current solution to identify better configurations. Starting from an initial solution, the heuristic evaluates nearby solutions by making incremental changes, such as swapping the positions of two boxes or moving a box to a different position (Scholz & Wäscher, 2017). If a neighbour solution has a lower completion time, it replaces the current solution. This iterative process continues until no further improvements can be identified, although it may sometimes get stuck in local optima. While local search is advantageous due to its simplicity and quick convergence, its primary limitation is the tendency to get trapped in local optima, potentially missing the best overall solution.

Simulated Annealing

Simulated annealing stands out as a particularly effective improvement heuristic. This method allows for occasional acceptance of worse solutions to escape local optima, with the probability of acceptance decreasing over time (Atmaca & Ozturk, 2013). Beginning with an initial solution and a high "temperature," the heuristic makes more random adjustments to the solution. If the new solution has a lower completion time, it is accepted. If not, it may still be accepted based on a probability that decreases as the temperature lowers. This enables the exploration of a wide solution space while gradually focusing on the most promising areas, aiming to find a global optimum.

Simulated annealing is especially beneficial for large and complex problems due to its balance of exploration and exploitation of the solution space. Its ability to accept worse solutions temporarily allows it to escape local optima and move towards a global optimum. This makes simulated annealing suitable for the reallocation of items in Kardex shuttles, where avoiding local optima is crucial for achieving the best possible reallocation order. The primary challenge with simulated annealing is selecting appropriate parameters, such as the cooling schedule and initial temperature, but its robustness and flexibility make it a solid choice compared to other heuristics.

Tabu Search

Tabu search is another improvement heuristic that uses memory structures to avoid cycling back to previously visited solutions. Starting with an initial solution, it explores the neighbourhood by making changes such as moving a box to a different shuttle (Henn & Wäscher, 2012). To prevent revisiting the same solutions, a tabu list records recent moves and prohibits reversing them for a specified number of iterations. By systematically exploring the solution space and avoiding cycles, tabu search aims to find better solutions and escape local optima. While effective at thorough exploration, it requires careful management of the tabu list and parameter tuning to balance exploration and exploitation.

Variable Neighbourhood Search

Variable neighbourhood search (VNS) systematically alters the neighbourhood structure during the search process. Starting with an initial solution, VNS explores progressively larger neighbourhoods by applying different types of moves, such as swapping multiple boxes or shifting entire sections (Menéndez et al., 2017). If an improvement is found, the search returns to the smallest neighbourhood and continues. This method helps escape local optima by exploring a variety of solution spaces. While VNS is flexible and effective in escaping local optima, it requires careful tuning of neighbourhood sizes and move strategies.

3.6 Conclusion

In conclusion, several constructive heuristics and improvements are considered in the literature review. The constructive heuristics that are most appealing to use for the reallocation model are the nearest neighbour and farthest neighbour heuristics. This is because these heuristics are working in the opposite direction and are straightforward to use for warehousing problems. Furthermore, they tend to provide a good initial solution. While various improvement heuristics offer unique advantages, simulated annealing emerges as the most robust and flexible option. Its ability to balance exploration and exploitation, combined with its mechanism for escaping local optima, makes it particularly well-suited for complex optimization problems. By utilizing simulated annealing, the process of reallocating boxes can be iteratively refined, leading to a more efficient and effective allocation strategy that minimizes completion time and enhances overall operational efficiency. Finally, after conducting a broad literature review no publications could be found on decreasing volume in a VLM by a box allocation method and this research will fill this research gap.

Chapter 4 – Solution Design

This chapter provides the solution design and answers the following research questions:

RQ 5: " What would be an efficient box division model for Benchmark? "

RQ 6: " How could the costs of execution reallocations be determined by the reallocation model "

RQ 7: " Which KPIs can be used to measure the change in storage efficiency due to the implementation of the box allocation method for Benchmark? "

Section 4.1 creates an overview of the required steps of the solution design. Section 4.2 discusses the development of the box division model. Section 4.3 shows the creation of the reallocation model.

4.1 Overview of the Solution Design

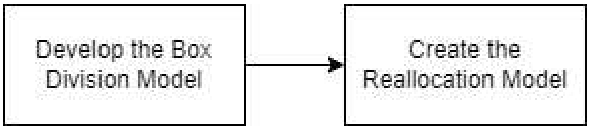

Figure 22: Overview of solution design steps for creating a box allocation method for Benchmark

As discussed in Section 1.4, this research will provide two deliverables for the box allocation method: the box division model and the reallocation model. Figure 22 shows the order in which these two models are developed. First, the box division model is created to indicate the recommended box type for the supplied SKUs and propose reallocations for boxes placed in the Kardex shuttles. After creating the box division model, the reallocation model is developed to determine the costs of executing the reallocations from the box division model. The goal of the reallocation model is to minimize the completion time for executing the reallocations coming from the box division model which also minimizes the costs of executing all reallocations.

4.2 Box Division Model

The first step in the solution design is developing the box division model. The box division model is developed as a Python application. This section aims to provide insights into the methods used by the box division model. Figure 23 provides the four steps that are performed in the box division model.

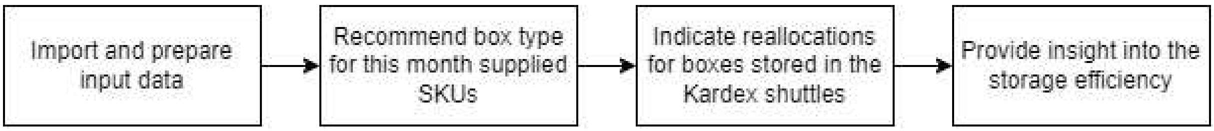

Figure 23: Overview of the Box Division Model

Import and Prepare Input Data

First, the box division model collects input data that will be used throughout all calculations. This input data is imported from the output files generated in the Kardex shuttles SKU analysis in Section 2.2.

Recommend Box Type for this Month's Supplied SKUs

Second, the box division model determines the recommended box type for SKUs supplied this month to prevent inefficient box type choices.

Indicate Reallocations for Boxes in the Kardex Shuttles
After the recommended box type is determined for supplied SKUs of this month, the box division model searches for inefficient box type choices in the Kardex shuttles that can be resolved. The box division model proposes reallocations based on three different reallocation methods that improve storage efficiency.

Provide Insight into the Storage Efficiency
Finally, the box division model provides insights into the storage efficiency of the Kardex shuttles before and after executing the proposed reallocations. Several KPIs are formulated that represent the storage efficiency of the Kardex shuttles at Benchmark.

4.2.1 Importing & Preparing Input Data
The first step in the box division model is to import and prepare data for calculations throughout the model. Figure 24 shows an overview of the processes that are happening in this step.

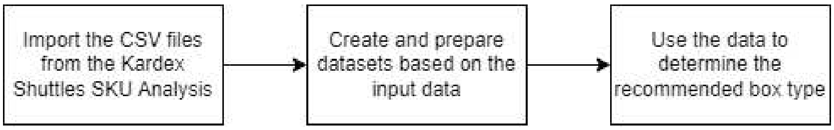

Figure 24: Overview of Importing Input Data

At first, the CSV files generated by the “Kardex SKU Analysis” as provided in Section 2.2 are imported by the box division model. During the import process, the files are saved as data frames in the Python application. These files include all information on SKUs stored in the Kardex shuttles. An important CSV file for the box division model is the “KardexInventoryAnalysis” as shown in Figure 20. This file contains an overview of the boxes stored in the Kardex shuttles and clearly shows all data surrounding the boxes.

4.2.2 Recommend Box Type for this Month’s Supplied SKUs
The second step in the box division model determines the recommended box type for this month's supplied SKUs. Figure 25 shows an overview of the processes in this step.

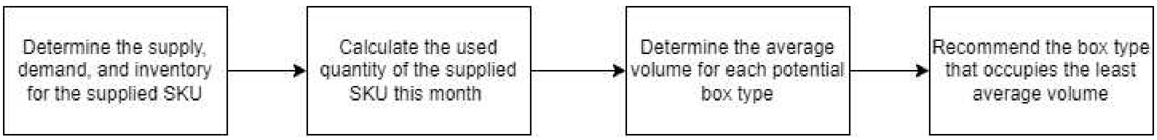

Figure 25: Overview determining the recommended box type for supplied SKUs

Determine the Supply, Demand & Current Inventory for Supplied SKUs
First, the supplied quantity, monthly demand, and current inventory of all supplied SKUs are determined based on the imported input data. The total demand is provided in the ”DemandAnalysis” file as shown in Figure 16 and the supplied quantity is included in the “SupplyAnalysis” file as provided in Figure 18. Finally, to determine the current inventory of the supplied SKUs the “TotalSKUInventory” as introduced in Figure 20. The box division model collects this data for each supplied SKU to calculate the used quantity.

Calculate Used Quantity of the Supplied SKUs
After the data on supply, demand, and current inventory is collected for all supplied SKUs, the used quantity of the supplied SKUs is calculated. In case the total demand for an SKU is higher than the current inventory of that SKU, it can be concluded that the supplied SKU will be used this month. Otherwise, the supplied SKU will not be used this month. The used quantity of the supplied SKU is required when the box division model calculates the average volume for each box type.

Calculate the Average Volume per Box Type

The box division model calculates the average volume throughout the month that each box choice would imply for supplied SKUs referred to as the "AverageBoxVolume". To calculate the "AverageBoxVolume" for each box type, this research created the following formula:

$$AverageBoxVolume = \frac{InitialBoxVolume + FinalBoxVolume}{2}$$

The equation to calculate the "AverageBoxVolume" introduces the parameters "InitialBoxVolume" and "FinalBoxVolume". "InitialBoxVolume" refers to the volume occupied when storing the supplied SKU in one or more boxes of a box type in the Kardex shuttles at the beginning of the month. "FinalBoxVolume" refers to the volume that the boxes occupy at the end of the month after all demand has occurred and some boxes might have been removed from the Kardex shuttles. The parameters are calculated using the following equations.

$$InitialBoxVolume = [\frac{SKUSuppliedQuantity}{SKUBoxQuantity}] * BoxTypeVolume$$

$$FinalBoxVolume = [\frac{RemainingSKUSuppliedQuantity}{SKUBoxQuantity}] * BoxTypeVolume$$

In the equation for the "InitialBoxVolume" there is divided the quantity of the supplied SKU "SKUSuppliedQuantity" by the quantity that fits in a box type "SKUBoxQuantity". This number is rounded up to the nearest integer indicated by the square brackets. The "SKUBoxQuantity" is determined by using the "SKU Bin Quantity" file as shown in Figure 19. Dividing "SKUSuppliedQuantity" by "SKUBoxQuantity" essentially provides the required number of boxes by choosing a specific box type. This number can be multiplied by the volume of a box type "BoxTypeVolume" to determine the "InitialBoxVolume" of the supplied SKU. As discussed, this process is repeated for all possible box types that could be used for the supplied SKU.

The "FinalBoxVolume" can be calculated similarly to the "InitialBoxVolume". The difference between the calculations is that "FinalBoxVolume" considers the remaining quantity of the supplied SKU at the end of the month indicated by the parameter "RemainingSKUSuppliedQuantity". This is calculated by subtracting the used quantity as calculated in the previous step of the box division model by the "SKUSuppliedQuantity" for each SKU. The "RemainingSKUSuppliedQuantity" is again divided by the "SKUBoxQuantity" to provide the remaining number of boxes that contain the supplied SKU at the end of the month. Multiplying this by the "BoxTypeVolume" provides the "FinalBoxVolume" after which the "AverageBoxVolume" can be calculated. Figure 26 shows an example of the "AverageBoxVolume" calculation for a supplied SKU. This example shows the "AverageBoxVolume" for each possible box type. It can be seen that the lowest "AverageBoxVolume" is obtained by choosing 10 B001 boxes when the supplied SKU arrives at Benchmark leading to an "AverageBoxVolume" of 26.3L.

Input Values	
TotalSKUInventory	20
SKUSuppliedQuantity	40
Demand	40

Box Type	BoxTypeVolume	SKUBoxQuantity	Initial Boxes	Final Boxes	InitialBoxVolume	FinalBoxVolume	Average Volume
B001	3.5	4	10	5	35	17.5	26.3
B002	10.2	8	5	3	51	30.6	40.8
B003	26.4	16	3	2	79.2	52.8	66.0
B004	76.8	20	2	1	153.6	76.8	115.2
B005	28.8	10	4	2	115.2	57.6	86.4
B006	4	2	20	10	80	40	60.0

Figure 26: Example of the Average Volume Calculation for a supplied SKU

Recommend Box Type

Figure 26 showed an example of calculating the "AverageBoxVolume" for one supplied SKU. In the Python application, this calculation is performed for all supplied SKUs and a recommended box type is determined including the number of boxes required. Figure 27 shows an overview of a couple of SKUs that are supplied at Benchmark. The box division model has determined the recommended box type and listed the " RecommendedBoxType" and "RequiredBoxes" in the data frame named "Supply Data".

Index	SKU	Supply	Due Date	Rank	Full Date	Demand	RecommendedBoxType	RequiredBoxes	AverageBoxVolume
0	ASM0051_240_00571-P	15	9	1	2024-09-24 00:00:00	22	B003	3	79.2
1	ASM0051_240_01792-LF	10	9	1	2024-09-20 00:00:00	13	B002	1	10.2
2	ASM0051_240_02281-LF	6	9	1	2024-09-24 00:00:00	14	B005	1	28.8
3	ASM0051_240_02542-P	1	9	1	2024-09-13 00:00:00	1	B004	1	76.8
4	ASM0051_240_03181-P	5	9	1	2024-09-20 00:00:00	12	B003	1	26.4

Figure 27: Overview of the "Supply Data" data frame created by the box division model

4.2.3 Indicate Reallocations for Boxes Stored in the Kardex Shuttles

The third step in the box division model is to indicate reallocations for boxes stored in the Kardex shuttles. Reallocations change the box division in the Kardex shuttles to improve storage efficiency. The box division model introduces three reallocation methods: One-to-One Reallocations, One-to-Many Reallocation, and Carrier Type Changes. At first, this section discusses the calculation for the recommended box type of boxes stored in the Kardex shuttles which will imply the One-To-One and One-to-Many reallocations.

Recommended Box Type for Boxes Stored in Kardex Shuttles

When calculating a recommended box type for boxes stored in the Kardex shuttles, the box division model uses a similar approach to the supplied SKUs. The only key difference is that now there is looked at the quantity of an SKU stored in a box at the beginning of the month "CurrentBoxQuantity" and the remaining quantity of an SKU in a box at the end of the month "RemainingBoxQuanity".

$$InitialBoxVolume = [\frac{CurrentBoxQuantity}{SKUBoxQuantity}] * BoxTypeVolume$$

$$FinalBoxVolume = [\frac{RemainingBoxQuantity}{SKUBoxQuantity}] * BoxTypeVolume$$

The "CurrentBoxQuantity" is provided for each box by the "KardexInventoryAnalysis" as shown in Figure 20. The "RemainingBoxQuanity" is determined by subtracting the used quantity of the box from the quantity stored in the box. The used quantity can be determined using the demand for an SKU, the current box quantities of boxes storing that SKU, and the rank of the boxes. As discussed, the box division model uses a rank for each box in inventory to see which box will be used first if the same SKU is stored at multiple locations. Therefore, the boxes with the lowest rank will be emptied first. Figure 28 shows an example of the calculation of the "RemainingBoxQuanity" for an SKU stored in two different boxes with a different rank.

```
SKUDemand =15       # Demand of this month

Box1.CurrentBoxQuantity = 10
Box2.CurrentBoxQuantity = 10

Box1.Rank = 1       # Lowest rank, so will leave first
Box2.Rank = 2       # Highest rank, so will leave last

Box1.RemainingBoxQuantity = 0  # All 10 required to fulfil the SKUDemand
Box2.RemainingBoxQuantity = 5  # Only 5 required to fulfil the SKUDemand
```

Figure 28: Example of the "RemainingBoxQuantity" calculation

Hence, for each possible box type the “InitialBoxVolume” and “FinalBoxVolume” are determined after which the “AverageBoxVolume” can be calculated for each box type. Similar to the recommended box type for supplied SKUs, the “AverageBoxVolume” is calculated using the formula as introduced in Section 4.2.2. All possible alternative box type choices are considered in search of the lowest “AverageBoxVolume”. The box type with the least “AverageBoxVolume” will be chosen as the recommended box type.

One-to-One Reallocation

If the recommended box type requires changing one box to a single box of a different type, a One-to-One (OTO) reallocation is performed. Figure 29 shows a visualization of an OTO reallocation of an SKU currently stored in a B004 that will be put into a B003 box. As can be seen in Figure 29, changing boxes instantly reduces the volume used to store the SKU from 76.8 to 26.4 Liter. The choice for the new location containing an unoccupied box of the required type is an action that needs to be performed by employees at the Kardex shuttles and is not automated via the PPG software. This choice of finding a location by the employees for the unoccupied required box type coming from OTO reallocations can be scheduled as will be shown in the reallocation model in Section 4.3.

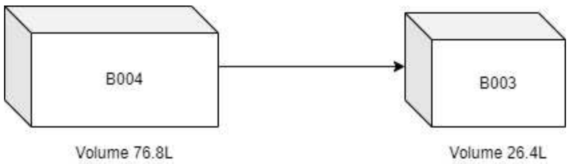

Figure 29: Example of a One-to-One Reallocation

One-to-Many Reallocation

If the recommended box type requires one box to be changed by two or more boxes a One-to-Many (OTM) reallocation is performed. Figure 30 shows an example of applying the OTM reallocation at the beginning of the month.

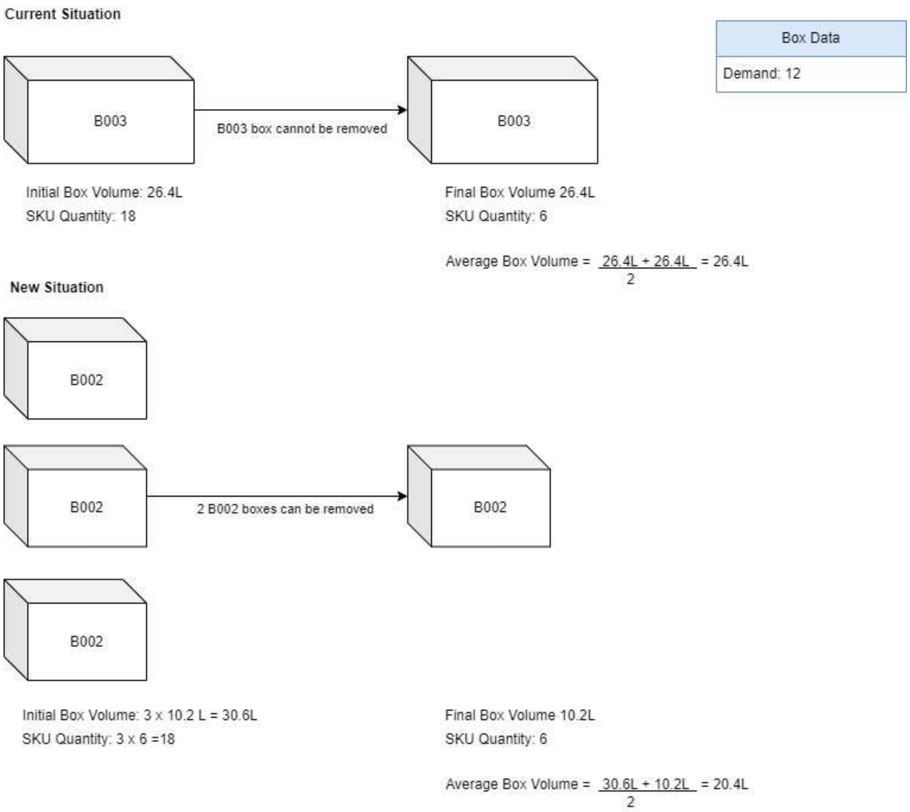

Figure 30: Example of a One-to-Many Reallocation

In the current situation, an SKU is stored in a single B003 box having an “Initial Box Volume” of 26.4 L at the beginning of the month. From the available box data, this month’s demand for the SKU stored in the box is 12. The SKU quantity stored in the B003 box is 18 which means that at the end of the month, there is still a quantity of 6 remaining in the B003 box meaning the box will not leave the Kardex shuttle. Therefore, the “Final Box Volume ” and “Average Box Volume” are equal at 26.4 L.

In the new situation, an OTM reallocation is performed by reallocating one single B003 box containing 18 items to three B002 boxes containing 6 items each. This increases the “Initial Box Volume” to 30.6L at the beginning of the month compared to 26.4L in the current situation. However, due to the demand of 12, two of the three B002 boxes can be removed this month. This leads to a “Final Box Volume” of 10.2L and an “Average Box Volume” of 20.4L. Since the “Average Box Volume” is lower in the new situation compared to the current situation, performing an OTM reallocation for the B002 box would be beneficial to save volume.

OTM reallocations require multiple boxes during the reallocation which is a more extensive operation to fulfil. Furthermore, OTM reallocations have to be performed via the PPG software which allocates boxes to random put locations containing unoccupied required boxes. This means that required boxes are placed at random put locations in the Kardex shuttles. Unfortunately, this random allocation by the PPG software cannot be changed. Therefore, the random allocation process should be modelled to determine the completion time for OTM reallocations in the reallocation model.

Carrier Type Change
Another reallocation method that could be used is the carrier type change. Carrier type changes can be performed to change the carrier type of some carriers that are placed in the Kardex shuttle as shown in Table 6. Table 10 shows for each carrier type which boxes could fit on this carrier type based on height and the boxes each carrier type contains. A special carrier type is the B006 carrier which not only contains 34 B006 boxes but also 11 B002 boxes. Furthermore, the B004 carrier type could contain any box type since it is the largest box in height as shown Table 4. Therefore, a B004 carrier could be changed to any other carrier type.

An example of using carrier type changes could be when the B003 box currently has a high occupancy rate and has almost no free boxes left in the Kardex shuttles. In this case, it could be possible to change a B004 carrier to become a B003 carrier. This way more storage capacity is created for the B003 box type by removing 10 B004 boxes and replacing these with 20 B003 boxes on the B004 carrier. Hence, the B004 carrier now becomes of carrier type B003. Note that this is only possible if the B004 box has sufficient empty boxes available for this carrier type change. Also, by performing this carrier type change the occupancy rate of the B004 boxes will increase while the B003 occupancy rate decreases.

Table 10: Overview of possible carrier changes

Carrier Type	Box Types that fit on Carrier Type	Contains Boxes
B001	B001	80 B001 boxes
B002	B001, B002, B005,B006	40 B002 boxes
B003	B001, B002, B003, B005, B006	20 B003 boxes
B004	B001, B002, B003, B004, B005, B006	10 B004 boxes
B005	B001, B005	10 B005 boxes
B006	B001, B002, B005, B006	34 B006 & 11 B002 boxes

Finally, it is efficient to perform carrier type changes on the least occupied carriers. In case there are currently occupied boxes on a removed carrier, these must be reallocated to a different carrier using OTO reallocation. Therefore, the removal of occupied boxes on the least removed occupied carriers will be considered in the reallocation model as will be discussed in Section 4.3.2.

Key Parameters

Figure 31 shows the parameters in the box division model that can be entered by Benchmark.

```
# Key Parameters
MaxOccupancy = 0.90

VolumeSavingsOTO = True
VolumeSavingsOTM = True
AllowCarrierChanges = True
```

Figure 31: Overview of Key Parameters that should be set in the Box Division Model

MaxOccupancy is a parameter included to set the desired maximum occupancy rate per box type at the end of the month. Since the OTO and OTM reallocations must be executed and these change box types, they directly influence the occupancy rates of the box types. It could be the case that a certain box type e.g. B003 is frequently chosen as the recommended box type by the OTO and OTM reallocation. Therefore, it could be that the occupancy rate of the B003 could go above 100%. By using a value below or equal to 100% for MaxOccupancy, the box division model will make sure that carrier type changes are performed such that the occupancy rate for each box type will not exceed the MaxOccupancy. Moreover, by lowering the value for MaxOccupancy the box division model could use the unoccupied space of the box types with a low occupancy rate and use this for box types with a high occupancy rate. Therefore, a more balanced occupancy rate is created for box types by lowering the MaxOccupancy instead of having large differences in high and low occupancy rates for box types in the Kardex shuttles. This is possible since carrier type changes can create additional boxes for box types facing a high occupancy rate by removing boxes of box types that have a lower occupancy rate.

Finally, three switches have been included that can be set to True or False for the reallocation methods If a reallocation method is set to True it will be used by the box division model. On the other hand, if a reallocation method is set to False it will not be used.

4.2.4 Provide Insight into the Storage Efficiency

The last step in the box division model is to provide insight into the storage efficiency of the Kardex shuttles. Several KPIs have been created to provide this insight and measure the impact of using the reallocations proposed by the box division model. Furthermore, the number of boxes of each type throughout the next month is visualized by the box division model.

Kardex Shuttle Volume KPIs

The box division model uses several KPIs to clearly show the impact on storage efficiency in the Kardex shuttles. These KPIs are tailored towards the volume usage in the Kardex shuttles and clearly show how Benchmark uses the *TotalStorageVolume* of the Kardex shuttles as calculated in Section 2.1.4. Table 11 shows several KPIs created for the box division model including a description.

Table 11: Overview of the output parameters of the box division model

KPI	Definition
OccupiedVolume	OccupiedVolume shows all volume that is occupied by the carriers and boxes in the Kardex shuttles (as shown in Section 2.1.4)
UnoccupiedVolume	UnoccupiedBoxVolume shows all volume that is unoccupied in the Kardex shuttles and could be used for placement of additional carriers and boxes (as shown in Section 2.1.4)
TotalVolumeOccupancy	TotalVolumeOccupancy is dividing the OccupiedVolume by TotalStorageVolume to indicate the percentage of volume used in the Kardex shuttles (using TotalStorageVolume from Section 2.1.4)
TotalMaterialCost	This KPI shows the total material costs caused by executing the box division model and purchasing additional boxes.

The KPIs are useful for executing experiments and comparing the values between experiments. The box division model provides an overview of the four KPIs before and after the reallocations. This allows us to compare the impact before and after implementing the reallocations proposed by the box division model for Benchmark. The TotalMaterialCost depend on the amount of boxes that need to be purchased of each type due to the reallocations. Table 4 showed the net purchase price and available boxes of each type. Since several box types still have available boxes these will be used first before ordering additional boxes. In case additional boxes are required these are ordered and will lead to TotalMaterialCost based on the quantity needed and the purchase price of each box type.

Forecasting the Maximum Boxes per Type

After showing the impact of using the box division model, a forecast on the number of boxes per box type stored in the Kardex shuttles is created. This forecast shows the number of boxes available for each box type throughout every day of the month has been determined to gain insight into the box division throughout the month for the Kardex shuttles. In this calculation, the supply is allocated into the recommended box type on the proposed delivery date and all reallocations have been performed at the beginning of the month. Also, the demand for the SKUs is used to calculate the number of boxes per box type that leave the Kardex shuttles on each day of the month based on the FIFO policy. Therefore, there can be calculated the number of boxes remaining on each day as shown in Figure 32.

Figure 32: Visualization of the number of boxes throughout the month after using box division model

Proposed Reallocations

After the box division model is executed it provides an overview of the proposed actions. These actions are coming from the OTO reallocations, OTM reallocations, and Carrier Type Changes. Figure 33 shows an impression of the proposed actions for each of the three reallocation methods by the box division model as displayed in the Python application. Furthermore, the box division model exports a detailed overview of the OTO and OTM reallocations that are suggested. An impression of these files is illustrated in Appendix G.

```
############################   Actions    #################################
The following actions need to be performed by One to One reallocation
0 B001 boxes have to be reallocated
197 B002 boxes have to be reallocated
202 B003 boxes have to be reallocated
134 B004 boxes have to be reallocated
54 B005 boxes have to be reallocated
2 B006 boxes have to be reallocated

The following actions need to be performed by One to Many reallocation
0 B001 boxes have to be reallocated
204 B002 boxes have to be reallocated
371 B003 boxes have to be reallocated
218 B004 boxes have to be reallocated
24 B005 boxes have to be reallocated
0 B006 boxes have to be reallocated

The following changes have been made for the carriers
B001: No changes made
B002: 8 Carriers Added
B003: 13 Carriers Added
B004: 38 Carriers Removed
B005: 8 Carriers Added
B006: 9 Carriers Added
```

Figure 33: Example of proposed actions by the box division model

4.3 Reallocation Model

The reallocation model has been developed to calculate the costs and completion time of executing the reallocations proposed by the box division model. Figure 34 shows an overview of the steps required to perform all reallocations which all imply labour costs based on the time of completing these steps.

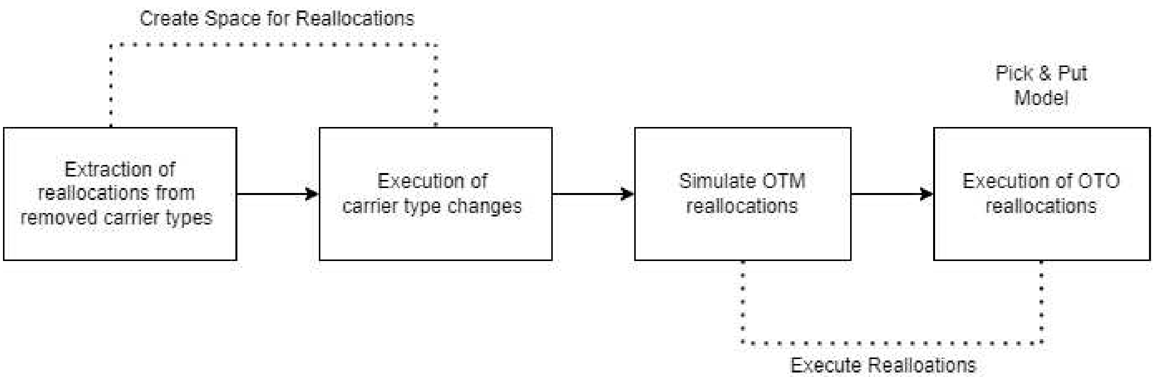

Figure 34: Process Flow of the Reallocation Model

Extraction of Reallocations from Removed Carrier Types

The first step in the reallocation model extracts boxes from the Kardex shuttles that need to be reallocated through either OTO or OTM reallocation and are placed on removed carrier types. The removed carrier types are indicated by carrier type changes and provided in the proposed actions of the box division model as illustrated in Figure 33. In case of Figure 33, the only removed carrier type would be B004 where 38 B004 carriers should get a different carrier type.

This step is necessary because the second step of the reallocation model, which executes carrier type changes, creates OTO reallocations for occupied boxes on removed carriers to unremoved carriers of the same carrier type. For example, if a B004 carrier gets a carrier type change, the occupied boxes on this carrier must be reallocated via OTO reallocations to other B004 carriers that are not removed and have space for these boxes. In case these occupied B004 boxes were also scheduled for an OTO or OTM reallocation, there are performed two reallocations instead of one. Therefore, extracting reallocations from removed carrier types prevents unnecessary OTO reallocations.

Execution of Proposed Carrier Type Changes
In the second step, specific carriers are selected for carrier type changes to obtain the desired carrier layout for the Kardex shuttles. All proposed carrier type changes indicated by the box division model in Figure 33 should be executed to create space for the OTO and OTM reallocations. Without performing the carrier type changes in the second step it could be that there is insufficient space for the OTO and OTM reallocations that have to be performed. Therefore, the carrier type changes make sure that all OTO and OTM reallocations can be executed.

Simulate OTM Reallocations
The third step of the reallocation model simulates the execution of the OTM reallocation. As discussed in Section 4.2.3, the PPG software performs the OTM reallocations by itself without any possibility for manual input or optimization. PPG uses random put location selection for OTM reallocation which is simulated in this step to get an impression of the time of completion and cost of executing OTM reallocations.

Execution of OTO Reallocations
The last step calculates the completion time of executing all OTO reallocations. The OTO reallocations are scheduled based on a mathematical model. Solutions to the mathematical model are found using several heuristic approaches.

Table 12: Overview of dedicated shuttles for order pickers

Order Picker	Shuttles
Order Picker 1	SH01, SH02
Order Picker 2	SH03, SH04
Order Picker 3	SH05, SH06
Order Picker 4	SH07, SH08
Order Picker 5	SH09, SH10

Benchmark addresses that they want to perform all steps with five internally active order pickers in their current way of working. This means the current method for order pickers to work at two shuttles in parallel will be maintained as introduced in Section 1.2.4. However, this way of working can only be used in the first and second steps of the reallocation model. This is because in the third step, PPG allocates random put locations to the OTM reallocations and the fourth step uses heuristics to assign reallocations containing pick and put tasks for all OTO reallocations divided over the whole warehouse. Therefore, the third and fourth steps require more flexibility than the first two steps and no dedicated shuttles can be used. Table 12 shows the overview of the dedicated shuttles allocated to the order pickers in the first and second steps of the reallocation model. Note that the order pickers have an internal cost rate of 60 euros per hour or one euro per minute.

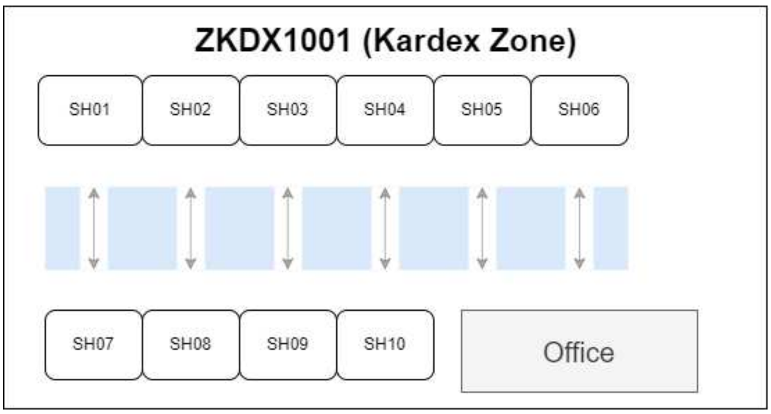

Figure 35: Overview of the temporary ZKDX1001 zone

Figure 35 shows an impression of the temporary layout of the Kardex zone during the execution of the reallocations. The blue zones contain pallets used to temporarily store boxes during the first and second steps of the reallocation model. Also, walking lanes are indicated using arrows between the shuttles. This is done since boxes are extracted from the Kardex shuttles in the first and second steps which must be stored outside of the Kardex shuttles before being executed in the third and fourth steps. These extracted reallocations will be referred to as the extracted OTO and extracted OTM reallocations.

4.3.1 Extraction of Reallocations from Removed Carrier Types

Figure 36 shows an overview of the first step in the reallocation model as provided in Figure 34. As explained, this step schedules the extraction of reallocations from removed carrier types referred to as the extracted OTO or OTM reallocations. The reallocation model is developed using a Python application and a detailed pseudocode on the used logic for the first step is provided in Appendix H.

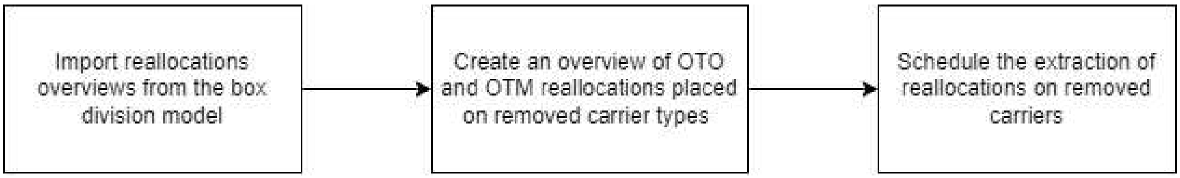

Figure 36: Overview of scheduling the extractions of reallocations from removed carrier types

Import Reallocation Overviews from the Box Division Model

At first, the reallocation overviews provided by the box division model are imported as illustrated in Appendix G. These overviews contain all OTO reallocations and OTM reallocations. Furthermore, in the Python application, the requested carrier type changes are entered as proposed by the box division model. An impression of this input section in the Python application is provided in Figure 37.

```
# Changes needed for carriers
CarrierChanges = {
    "B001": 0,    # No changes made
    "B002": 5,    # Carriers added
    "B003": 15,   # Carriers added
    "B004": -35,  # Carriers removed
    "B005": 7,    # Carriers added
    "B006": 8     # Carriers added
}
```

Figure 37: Overview of input section for the requested carrier type changes

Create Overview of Reallocations on Removed Carrier Types

After importing the reallocations coming from the three reallocation methods by the box division model, an overview of which reallocations should be extracted from the Kardex shuttles is created. First, there is looked at the removed carrier types and created an overview of all OTO and OTM reallocations placed on these carrier types. In case of the proposed carrier type changes from Figure 37, this would mean that the reallocation model searches for all OTO and OTM reallocations that are located on carrier type B004. This overview contains for each shuttle the locations that contain reallocations that should be extracted

Extracting Reallocations from Removed Carrier Types

As shown in Table 12, each order picker has two dedicated shuttles for the first step of the reallocation model. Since the order pickers have two dedicated shuttles they are extracting all OTO and OTM reallocations that should be extracted from the two shuttles. To calculate the picking time of extracted OTO and OTM reallocations two different formulas are used depending if reallocations are still present on both Kardex shuttles assigned to the order picker. In case reallocations are still present at two shuttles the following formula is to calculate the picking time of these reallocations:

$$PickingTime = PurgeTime + TempStoreTime = 10 + 30 = 40\ sec$$

In this equation, *PurgeTime* refers to the time it takes to digitally realize the pick action determined at 10 seconds, while *TempStoreTime* determined at 30 seconds indicates the time it takes to physically extract a box and place it in a temporary box on the pallets in the blue zones shown in Figure 35. Note that when using two shuttles simultaneously there is no waiting time for requesting carriers and *CarrierTravelTime* is excluded as explained in Section 1.2.4.

In case only one shuttle contains reallocations that must be extracted, the advantage of using two shuttles and excluding *CarrierTravelTime* is lost. *CarrierTravelTime* is determined to be 25 seconds, calculated by taking the mean time of completion for executing 10 picks on a single Kardex shuttle at different carriers. Therefore, the formula to calculate picking time in case only one shuttle contains reallocations that must be extracted is as follows:

$$PickingTime = PurgeTime + TempStoreTime + CarrierTravel\ Time = 10 + 30 + 25 = 65\ \text{sec}$$

As can be seen in the pseudocode stored in Appendix H, order pickers are assigned reallocations on different shuttles using the parallel workflow until only one shuttle contains reallocations that must be extracted. Moreover, the picking time calculation differs in both situations. Finally, after all reallocations have been assigned and performed a printed schedule is generated including a summary of the time of completion and total costs of the first step in the reallocation model.

4.3.2 Execution of Proposed Carrier Type Changes

As shown in Figure 34, the second step of the reallocation model should schedule the execution of the proposed carrier type changes coming from the box division model. Figure 38 shows an overview of the steps taken during this step in the reallocation model and Appendix I provides a detailed pseudocode about the methods used for this step in the Python application.

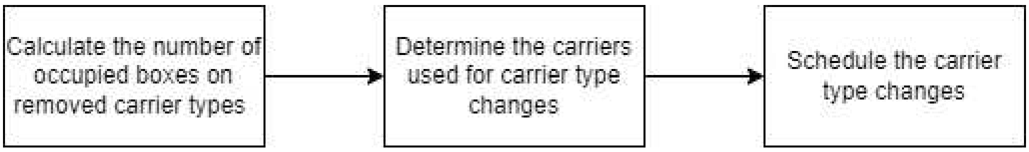

Figure 38: Overview of the logic for the carrier change process

Calculate the Number of Occupied Boxes on Removed Carrier Types
First, an overview of the number of occupied boxes on the removed carrier types is created. This is done to gain insight into the least occupied carriers on removed carrier types. By determining the least occupied carriers there can be effectively chosen which carriers should be removed from the Kardex shuttles for the carrier type changes.

Determine the Carriers Used for Carrier Type Changes
After an overview of the number of occupied boxes on the removed carrier types is calculated, the least occupied carrier type changes are indicated for removal. This is done since removing the least occupied carriers results in the least amount of OTO reallocations. The OTO reallocations on removed carriers are extracted and placed in temporary boxes in the blue zones as shown in Figure 35. Therefore, these removed OTO reallocations are also referred to as extracted OTO reallocations.

Schedule the Carrier Type Changes

After indicating the removed carriers for realizing the carrier type changes, these can be executed by the order pickers and are scheduled by the reallocation model. As discussed, throughout the second step of the reallocation model the order pickers are still working on the two dedicated shuttles as defined in Table 12. As can be seen in the pseudocode in Appendix I, there is again scheduled using the parallel workflow. To calculate the time of performing a carrier type change for a carrier the following formula is used in the second step of the reallocation model:

$$CarrierChangeTime = CarrierTravelTime + RemovalTime + AdditionTime$$

CarrierChangeTime uses three main parameters which are *CarrierTravelTime*, *RemovalTime*, and *AdditionTime*. These components are using the following equations in the model. Note that the completion time in seconds related to a variable is placed behind the variables in brackets.

$$CarrierTravelTime = \begin{cases} 25, & if\ there\ are\ only\ jobs\ available\ at\ one\ shuttle \\ 0, & if\ jobs\ are\ available\ at\ two\ shutlles\ or\ at\ the\ same\ carrier \end{cases}$$

$$\begin{aligned} RemovalTime = {} & OccupiedBoxes * \left(PurgeTime\ (10) + TempStoreTime\ (35)\right) \\ & + ClearCarrierTime\ (10) + BoxesToRemove * EmptyBinRemovalTime(10) \end{aligned}$$

$$AdditionTime = FillCarrierTime\ (10) + BoxesToAdd * EmptyBinStoreTime(10)$$

Here, *CarrierTravelTime* is equal to value used in the first step as determined in Section 4.3.1. *RemovalTime* calculates the time it takes to remove all boxes on a removed carrier. For each occupied box it considers *PurgeTime* which is the time required *to* digitally realize the picking action and the *TempStoreTime* of picking items from the occupied box into a temporary similar box on the pallets in the blue zones. The use of temporary boxes on the pallets is required for extracting boxes since PPG can only clear carriers that are fully occupied with boxes. This means that after removing the items of an occupied box into a temporary box the empty box will be placed back on the removed carrier. After all occupied boxes are removed the *RemovalTime* takes into account the time it digitally takes to remove a complete carrier in PPG or the *ClearCarrierTime*. Finally, the time of removing all empty boxes is calculated by multiplying all *BoxesToRemove* on a carrier with the *EmptyBinRemovalTime*.

Finally, the *AdditionTime* calculates the time it takes to fill the carrier by multiplying the amount of *BoxesToAdd* with the *EmptyBinStoreTime* and adding up the required digital action in the PPG software *FillCarrierTime*. After scheduling all carrier type changes, the reallocation model provides a summary of the total completion time and costs. Furthermore, a visual representation of the job division between the order pickers is made.

4.3.3 Simulate Execution of OTM Reallocations

The third step of the reallocation model simulates the execution of the OTM reallocations as shown in Figure 34. As discussed, the PPG software randomly determines the put locations of the OTM reallocations and employees don't influence which locations are used for the put. Therefore, this random process is recreated in the reallocation model such that the completion time for the OTM reallocations and total costs still can be calculated. A detailed pseudocode of the methods used in the Python application for the third step of the reallocation model is shown in Appendix J. An overview of the steps taken in the third step of the reallocation model is provided in Figure 39.

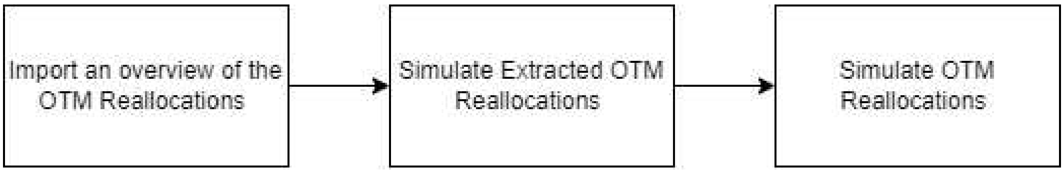

Figure 39: Overview of the steps in the simulation of OTM reallocations

Import an overview of the OTM reallocations

The first step in the simulation of the OTM reallocations is to import an overview of the OTM reallocations. These overviews contain the OTM reallocations proposed by the box division model as illustrated in Appendix G. Also, the extracted OTM reallocations which were placed on the pallets in the first step of the reallocation model must executed in this step. Therefore, both the extracted OTM reallocations and OTM reallocations that are still placed in the Kardex shuttles are executed in this step of the reallocation model.

Simulate Extracted OTM reallocations

The second step simulates the execution of the extracted OTM reallocations. As the pseudocode shows in Appendix J, the simulation is performed over 1000 runs. This is due to the randomness of the put location selection by PPG. Performing multiple runs will provide a better impression of the expected total completion time and costs. As discussed OTM reallocations split up one box into two or more boxes. However, there is a difference in calculating the time of completing an OTM reallocation when exactly two boxes are required for the recommended box type.

<u>OTMCompletionTime for exactly two required boxes</u>

To calculate the completion time for an OTM reallocation requiring exactly two boxes for the recommended box type the following equation can be used:

$$OTMCompletionTime = StorePrep + RemovalTime + AdditionTime$$

Here, *OTMCompletionTime* consists of three components for simulating extracted OTM reallocations:

$$StorePrep = 10 * RequiredBoxes = 10 * 2 = 20$$

StorePrep time represents the time it takes to generate put locations for the OTM reallocations using PPG digitally. It depends on the required boxes for the recommended box type, referred to as *RequiredBoxes*.

$$RemovalTime = TravelTime + BoxPickupTime[10,20]$$

RemovalTime indicates the time it takes to pick up the box that requires the OTM reallocation. The *RemovalTime* contains *TravelTime* and *BoxPickupTime*. The *TravelTime* depends on the current location and the location where the original box is placed. Furthermore, the *BoxPickupTime* has been included which is the time it takes to pick up the box from the location which varies between 10 and 20 seconds.

$$AdditionTime\ = TravelTime + \ CarrierTravelTime\ (25) + Put\ Time[10,20]$$

Lastly, the time to put the SKU in the new boxes is determined using *AdditionTime*. *AdditionTime* includes *TravelTime*, *CarrierTravelTime* and *PutTime*. The *TravelTime* is determined similarly to the approach used in the *RemovalTime*. However, it now considers the time between the pickup location and put location containing the required boxes for the OTM reallocation. *CarrierTravelTime* is again equal to the value of 25 as introduced earlier. Finally, *PutTime* varies between 10 to 20 seconds indicating the execution of several puts in the required boxes of the recommended box type.

OTMCompletionTime for more than two required boxes
To calculate the time it takes to execute an OTM reallocation involving more than two required boxes following equation is used:

$$OTMCompletionTime = \ PrintTime + StorePerp + \ RemovalTime + AdditionTime$$

The difference in comparison to the previously shown *OTMCompletionTime* function is that *PrintTime* has been introduced using the equation:

$$PrintTime = 30\ sec$$

PrintTime is introduced since extra labels need to be printed for the SKU information. Two labels are printed when performing any OTM reallocation, but now more than two labels are required and additional labels need to be printed.

Furthermore, the *PutTime* used in the *AdditionTime* has increased depending on the *RequiredBoxes* for the OTM reallocation.

$$PutTime = [5 * RequiredBoxes, 10 * RequiredBoxes]$$

As can be seen in the pseudocode of Appendix J, after scheduling all extracted OTM reallocations a summary of completion time and costs is created and a visual representation of a schedule for the order pickers is made.

Simulate OTM reallocations
For the extracted OTM reallocations, a PurgeOrder has already been performed in PPG in the first step of the reallocations model and *PurgeTime* is not considered in the *OTMCompletionTime*. However, for OTM reallocations that are still in the Kardex shuttles, a PurgeOrder must be created to digitally realize the picking action of the OTM reallocations. Therefore, *PurgeTime* is included in the picking time formulas for simulating OTM reallocations.

$$PurgeTime = \ 10$$

The *PurgeTime* is similar to the value of 10, as introduced in the previous sections. The formula for the *OTMCompletionTime* when simulating the execution of OTM reallocations requiring exactly two boxes becomes:

$$OTMCompletionTime = \ PurgeTime + StorePrep + \ RemovalTime + AdditionTime$$

Finally, the formula for the *OTMCompletionTime* for simulating the execution of OTM reallocations requiring more than two boxes becomes:

$$OTMCompletionTime = \ PrintTime + \ PurgeTime + StorePerp + \ RemovalTime + AdditionTime$$

Again, there is created a summary of completion time and costs and a visual representation of the schedule.

4.3.4 Execution of OTO Reallocations

The fourth step in the reallocation model is to schedule the OTO reallocations as shown in Figure 34. The schedule for the OTO reallocations consists of pick and put actions for the order pickers. As discussed in Section 3.5, scheduling the OTO reallocations is a type of job shop scheduling and hence a combinatorial optimization problem. Therefore, heuristics are required to schedule the OTO reallocations. Based on the model of Ardjmand et al. (2018) introduced in Section 3.5.1, the following model is created for scheduling the OTO reallocations.

Model for Execution of OTO Reallocations

$\underline{Sets}$

$I = \{1,2, \dots, OTORecallocations\}$ $\quad$ $Set\ of\ OTO\ Reallocations\ to\ be\ performed$

$P = \{1, \dots, 5\}$ $\quad$ $Set\ of\ order\ pickers\ being\ active\ in\ the\ Kardex\ zone$

$B = \{B001, B002, B003, B004, B005, B006\}$ $\quad$ $Set\ of\ different\ box\ types$

$S = \{SH01, SH02, \dots, SH10\}$ $\quad$ $Set\ of\ Kardex\ shuttles$

$\underline{Indeces}$

$i, j \in I$ $\quad$ $Index\ for\ OTO\ reallocations$

$p \in P$ $\quad$ $Index\ for\ order\ pickers$

$b \in B$ $\quad$ $Index\ for\ the\ box\ types$

$s_1, s_2 \in S$ $\quad$ $Index\ for\ the\ shuttles$

$\underline{Parameters}$

$M_1(i) =$ $The\ machine\ (shuttle)\ where\ the\ \textbf{pick task}\ for\ reallocation\ i\ is\ located$

$tt(s_1, s_2) =$ $Travel\ time\ between\ shuttle\ s_1\ and\ shuttle\ s_2$

$StartLocation_{\ p} = Intital\ starting\ location\ of\ picker\ p$

$r\ (i) =$ $removal\ time\ in\ seconds\ for\ rellocation\ i$

$p\ (i) =$ $put\ time\ in\ seconds\ for\ rellocation\ i$

$d\ (i) =$ $digital\ OTO\ handling\ time\ in\ seconds\ for\ rellocation\ i$

$cw\ (i) =$ $waiting\ time\ when\ requesting\ a\ carrier\ for\ rellocation\ i$

$CurrentBoxType_i =$ $Current\ box\ type\ of\ of\ reallocations\ i$

$RequiredBoxType_i =$ $Required\ box\ type\ of\ reallocation\ i$

$CurrentBoxes_{b,s} =$ $Current\ number\ of\ boxes\ of\ type\ b\ placed\ in\ shuttle\ s$

$PickerCosts =$ $\text{Cost}\ rate\ of\ an\ order\ picker\ per\ \text{second}$

$\underline{Decision\ Variables}$

$$X_{i,\mathrm{p}} \quad \begin{cases} 1, & if\ reallocaton\ i\ is\ performed\ by\ order\ picker\ p \\ 0, & else \end{cases}$$

$$Z_{\mathrm{i,s,b}} \quad \begin{cases} 1, & if\ reallocation\ i\ is\ \textbf{put}\ at\ shutle\ s\ in\ box\ type\ b \\ 0, & else \end{cases}$$

$$Y_{i,\mathrm{j,p}} \quad \begin{cases} 1, & if\ reallocaton\ j\ follows\ task\ i\ for\ order\ picker\ p \\ 0, & else \end{cases}$$

$t_{i,p} =$ $Start\ time\ of\ reallocation\ i\ by\ order\ picker\ p$

$C_p =$ $Completion\ time\ for\ picker\ p$

Derived Equation

$M_2(i) = \sum_{s \in S} \sum_{b \in B} Z_{i,s,b} * s \quad \forall i$ *The put machine is determined based on the decision variable* $Z_{i,s,b}$

Objective Funciton

$$Min\ z = \sum_{p \in P} C_p * PickerCosts$$

The objective function minimizes the total costs of completing the reallocations.

Constraints

1. **Reallocation completeness**

$$\sum_{p \in P} X_{i,p} = 1 \quad \forall i$$

Each reallocation i must be performed by one order picker p

2. **Carrier and Shuttle Assignment**

$$\sum_{b \in B} \sum_{s \in S} Z_{i,s,b} = 1 \quad \forall i$$

$$\sum_{s \in S} Z_{i,s,RequiredBoxType_i} = 1 \quad \forall i$$

These constraints ensure that the put location for reallocation i is assigned to exactly one shuttle s and is put in the required box type.

3. **Flow Constraints**

$$\sum_{j \in I} Y_{i,j,p} = 1 \quad \forall i, \forall p, \ i \neq j,$$
$$\sum_{i \in I} Y_{i,j,p} = 1 \quad \forall j, \forall p, \ i \neq j$$

This constraint ensures that after executing reallocation i, another reallocation j must be executed.

4. **Capacity Constraints**

$$CurrentBoxes_{b,s} - \sum_{i \in I} Z_{i,s,b} \geq 0 \quad \forall b, \forall s$$

This constraint ensures that the assigned put shuttle has enough capacity for the required box types.

5. **First Reallocation Start Time**

$$t_{i,p} \geq tt\left(StartLocation_p, M_1(i)\right) \quad \forall i, \forall p$$

Before the first reallocation can be started by picker p, the picker must travel from its start location to the pick machine $M_1(i)$

6. **Subsequent Reallocations Start Time**

$$t_{j,p} \geq t_{i,p} + d\,(i) + cw\,(i) + r\,(i) + \sum_{b \in B} \sum_{s \in S} Z_{i,s,b} * tt\big(M_1(i), M_2(i)\big) + cw\,(i) + d\,(i) + p\,(i) + tt\big(M_2(i), M_1(j)\big) \quad \forall i, \forall j, \forall p, i \neq j$$

This constraint makes sure that the next reallocation performed by an order picker starts after the previous one completed

7. **Completion Time of Pickers Constraint**

$$C_p \geq t_{i,p} + d\,(i) + cw\,(i) + r\,(i) + \sum_{b \in B} \sum_{s \in S} Z_{i,s,b} * tt\big(M_1(i), M_2(i)\big) + cw\,(i) + d\,(i) + p\,(i) \quad \forall i, \forall p$$

This constraint makes sure that the completion times of the order pickers is updated correctly

8. **Different pick and put shuttle**

$$M_1(i) \neq M_2(i) \quad \forall i, \forall p$$

Make sure that the put shuttle is a different shuttle than the pick shuttle.

9. **Sign Constraints**

$X_{i,p,}$, $Y_{i,j,p}$, $Z_{i,,s}$ *Binary*
$t_{i,p}$, C_p *Interger*

The model described all sets, parameters, decision variables, and constraints. As discussed this model is created for Benchmark based on the job shop scheduling model of Ardjmand et al. (2018) and extended to make it suitable for Benchmark. To find a solution to this model heuristics are used and developed in a Python application.

Constructive Heuristics

As discussed in Section 3.6, the two constructive heuristics considered for this research are the nearest neighbour and farthest neighbour heuristics. In the Python application, there can be chosen between which heuristic is used for the reallocation model as can be seen in Figure 40.

```
# Setup
ConstructiveHeuristic = "NN"  # Choose between NN and FN
```

Figure 40: Option in the Python app to select a constructive heuristic

Nearest Neighbour Heuristic

The nearest neighbour heuristic aims to assign the closest feasible put location for the assigned reallocations of an order picker. To get a complete impression of the nearest neighbour heuristic a pseudocode is provided in Appendix K. Each reallocation has a pick location at which the SKU is stored in its current box type. The order picker has to visit this pick location to change the current box type to the required box type at the put location. The nearest neighbour heuristic chooses the closest shuttle with space for the required box type as the put location. To determine the closest distance there can be used the travel time between the possible pick and put shuttles as shown in Table 13.

Table 13: Overview of the travel time between the Kardex shuttles

Shuttle	SH01	SH02	SH03	SH04	SH05	SH06	SH07	SH08	SH09	SH10
SH01	0	3	6	9	12	15	3	6	9	12
SH02	3	0	3	6	9	12	6	3	6	9
SH03	6	3	0	3	6	9	9	6	3	6
SH04	9	6	3	0	3	6	12	9	6	3
SH05	12	9	6	3	0	3	15	12	9	6
SH06	15	12	9	6	3	0	18	15	12	9
SH07	3	6	9	12	15	18	0	3	6	9
SH08	6	3	6	9	12	15	3	0	3	6
SH09	9	6	3	6	9	12	6	3	0	3
SH10	12	9	6	3	6	9	9	6	3	0

<u>Farthest Neighbour Heuristic</u>

The farthest neighbour heuristic works in the opposite direction of the nearest neighbour heuristic. An impression of the farthest neighbour heuristic is provided with a pseudocode in Appendix M. It selects the farthest possible put location with space for the required box type of the reallocations assigned to an order picker. The farthest put location for the reallocations is again determined based on the travel time between the pick and put location as provided in Table 13.

Improvement Heuristics

After an initial solution is created using a constructive heuristic the improvement heuristic can be used to improve this solution. The improvement heuristic used for the reallocation model is the simulated annealing heuristic. Appendix N provides a pseudocode creating an overview of the simulated annealing heuristic used in the Python application. Simulated annealing changes the order of reallocations by creating neighbour solutions that can be accepted based on conditions. The order of reallocations is changed by using the move operator which can move reallocations to any different moment in time and to different order pickers. The following relations show when a solution will be accepted:

$$NeighborCost < CurrentCost$$
$$or$$
$$e^{\frac{(CurrentCost - NeighborCost)}{CurrentTemp}} > RandomNumberBetween\ [0,1]$$

As shown in these relations, a neighbour solution can be accepted if it improves the current solution. Another reason for accepting a neighbour solution is due to randomness in case it is not better than the current solution. As discussed in Section 3.5, simulated annealing uses randomness to allow diversification in the beginning but decreases the chance of accepting a worse neighbour solution over time the chance that a worse neighbour solution will be selected becomes smaller. The behaviour of the simulated annealing heuristic depends on several parameters as shown in Figure 41.

```
# Simulated Annealing Parameters
InitialTemp = 200
FinalTemp = 1
Alpha = 0.99
Iterations = 50
```

Figure 41: Parameters that can be set to influence the Simulated Annealing algorithm

Figure 41 shows the initial temperature, final temperature, alpha and number of iterations that can be adjusted when running the simulated annealing algorithm. The start and end temperature influences the amount of randomness that is allowed when exploring potential different solutions in combination with the alpha value. The current temperature starts equal to the initial temperature and decreases after performing the amount of iterations until reaching the final temperature. This is done by multiplying the alpha with the current temperature after performing a predetermined number of iterations as can be seen in the following equation:

$$CurrentTemp = CurrentTemp * Alpha\ (\alpha)$$

Due to this behaviour, in the beginning, a wider variety of solutions will be explored after which becomes more narrow and randomness decreases. The simulated annealing algorithm will stop if the current temperature becomes lower than the end temperature. After there is found an improved solution, a summary is provided of the total time and costs of the OTO reallocations that need to be performed. This is done for both the extracted OTO reallocations from Section 4.3.1 and the remaining OTO reallocations.

4.4 Conclusion

This chapter showed the development of the box division model and reallocation for Benchmark. For both models, there is clearly illustrated the methods and options. Using the box division model Benchmark is able to gain insight into the storage efficiency using several different KPIs. The box division model proposes reallocations based on three different reallocation methods that are aimed at improving the box type choices for boxes in the Kardex shuttles. Furthermore, it determines the recommended box type for supplied SKUs. The reallocation model can be used by Benchmark to gain insight into the total costs and time of completion for executing the reallocations proposed by the box division model. Therefore, using both models shows the impact on storage efficiency in comparison to the total costs of the implantation of the box allocation method at Benchmark.

Chapter 5 – Results and Evaluation

This chapter discusses the result of implanting a box allocation method at Benchmark and answers the following research question:

RQ 8: " What would be the impact on the storage efficiency by implementing the box allocation method at Benchmark? "

Section 5.1 discusses the output of the box division model. Section 5.2 provides the output of the reallocation model. Section 5.3 performs a cost analysis on implementing the box allocation method.

5.1 Output of the Box Division Model

This section shows the results of executing the box division model at Benchmark. The impact on the storage efficiency of the Kardex shuttles is measured for executing reallocations proposed by the box division model.

5.1.1 Required Actions

During the execution of the box division model the settings as shown in Figure 31 are used. These settings are used because Benchmark does not prefer an occupancy rate higher than 90% for each box type throughout the month. Therefore, the box division model uses all three reallocation methods and a MaxOccupancy of 0.90. However, to get an impression of the impact on storage efficiency using different values for the MaxOccupancy experiments are performed in Section 5.1.4.

Figure 42 shows the required actions coming from the box division model. For each reallocation method, there are provided several actions. For the OTO and OTM reallocations, the number of actions that must be performed for each box type is shown. The box division model also lists these OTM and OTO reallocations in output files providing more details as shown in Appendix G. Furthermore, for each carrier type there is listed the number of added or removed carriers due to the carrier type changes. The box division model removes 35 carriers only of the B004 carrier type. This can be explained since the B004 box is the largest box type and removing this box type has the most potential for volume savings.

```
############################   Actions    #################################
The following actions need to be performed by One to One reallocation
0 B001 boxes have to be reallocated
254 B002 boxes have to be reallocated
286 B003 boxes have to be reallocated
166 B004 boxes have to be reallocated
87 B005 boxes have to be reallocated
1 B006 boxes have to be reallocated

The following actions need to be performed by One to Many reallocation
0 B001 boxes have to be reallocated
248 B002 boxes have to be reallocated
402 B003 boxes have to be reallocated
281 B004 boxes have to be reallocated
43 B005 boxes have to be reallocated
0 B006 boxes have to be reallocated

The following changes have been made for the carriers
B001: No changes made
B002: 5 Carriers Added
B003: 15 Carriers Added
B004: 35 Carriers Removed
B005: 7 Carriers Added
B006: 8 Carriers Added
```

Figure 42: Overview of proposed actions by the box division model

5.1.2 Occupancy Changes

Figure 43 shows the overview of occupancy changes caused by reallocations of the box division model. At first, a comparison between the initial occupancy at the beginning of the month and the final occupancy at the end of the month can be seen for each box type. Benchmark does not exceed the desired occupancy rate of 90% at the end of the month. The B003 had an initial occupancy rate of 98.2% and this is now brought back to 89.6% as can be seen by the final occupancy. From these results, it can be observed that the box division model manages to stick to the MaxOccupancy. Furthermore, an overview of the initial occupied boxes, final occupied boxes, and maximum occupied boxes in a month is provided. Lastly, the difference in initial and final maximum boxes is shown which concerns the number of boxes placed of each box type in the Kardex shuttles.

```
###########################   Results    ################################
     Initial Occupancy Final Occupancy
B001            40.8%           61.2%
B002            78.2%           89.2%
B003            98.2%           89.6%
B004            82.3%           59.6%
B005            88.2%           87.7%
B006            72.4%           88.6%
      Initial Occupied Boxes  Final Occupied Boxes  Maximum Occupied in Month
B001                    2026                  3035                       3047
B002                    2477                  3084                       3126
B003                    3024                  3027                       3079
B004                     897                   441                        466
B005                     432                   491                        504
B006                     197                   482                        494
      Initial Maximum Boxes  Final Maximum Boxes
B001                   4960                 4960
B002                   3168                 3456
B003                   3080                 3380
B004                   1090                  740
B005                    490                  560
B006                    272                  544
```

Figure 43: Overview of the occupancy changes

5.1.3 Impact on Volume

Figure 44 provides an overview of the current situation without using the box division model and the improved situation using the box division model. In both situations, the KPIs Occupied Volume, Unoccupied Volume, and TotalVolumeOccupancy as formulated in Section 4.2.4 are calculated. Also, the total storage volume of all ten Kardex shuttles of 327.0 m3 is shown as determined in Section 2.1.4.

Figure 44 shows that using the box division model the occupied volume successfully decreased from 288.1 m3 in the current situation to 275.5 m3 in the improved situation. Furthermore, the box division model increases the unoccupied volume from 38.9 m3 in the current situation to 51.5 m3 in the improved situation, providing Benchmark with an additional 12.6 m3 of unoccupied volume. The TotalVolumeOccupancy decreases from 88.1% to 84.3%. Finally, the total material costs of all reallocations are determined at € 2131.46. The KPIs show that the box division model increases the storage efficiency in the Kardex shuttles.

```
############################   Volume Changes   #################################
Total Storage Volume of all 10 Kardex shuttles =  327.0 m3

Current Situation:
Occupied Volume =  288.1 m3
Unoccupied Volume =  38.9 m3
TotalVolumeOccupancy =  88.1 %

Improved Situation:
Occupied Volume =  275.5 m3
Unoccupied Volume =  51.5 m3
TotalVolumeOccupancy =  84.3 %

Total Material Costs of Actions: € 2131.46
```

Figure 44: Overview of volume changes by the box division model

5.1.4 Experiment Outcomes

This research also studies the improved situation when using different values than 90% for the Max Occupancy. Several experiments have been performed to see the impact of varying the MaxOccupancy on the KPIs. Table 14 shows the results of the performing the experiments:

Table 14: Experiment outcomes of the box division model

MaxOccupancy	Occupied Volume	Unoccupied Volume	Total Volume Occupancy	TotalMaterialCosts
90 %	275.5 m3	51.5 m3	84.3 %	€ 2131.46
89 %	274.6 m3	52.4 m3	84.0 %	€ 2330.06
88 %	273.4 m3	53.5 m3	83.6 %	€ 2831.18
87 %	272.5 m3	54.4 m3	83.4 %	€ 3508.38
86 %	271.1 m3	55.9 m3	82.9 %	€ 4306.28
85 %	269.6 m3	56.4 m3	82.4 %	€ 5462.08
84 %	-	-		-

As shown in Figure 43, the final occupancy rate of the B004 boxes is the lowest of all box types. By lowering the MaxOccupancy value in the experiments, B004 carriers can be used for different box types by performing carrier type changes as discussed in Section 4.2.3. This enables the occupancy rate of other box types to decrease by using unoccupied space from the B004 carriers to place additional boxes of different box types. Since the B004 is the largest box, removing even more B004 carriers for different box types would result in additional unoccupied volume in the Kardex shuttles. This is because all other box types occupy less volume than B004 box.

The experiments show that even more significant increases in the total unoccupied volume in the Kardex shuttles can be achieved. The highest increases can be achieved when using a MaxOccupancy of 85%. In this experiment, a maximum unoccupied volume of 56.4 m3 could be achieved implying a total material cost of € 5462.08. Note that it was not possible to achieve a MaxOccupancy lower than 85% for the Kardex shuttles.

These results show that Benchmark could increase the unoccupied volume even more by using different values for the MaxOccupancy lower than only 90%. Benchmark could use this overview of the experiments to determine which MaxOccupancy seems most appealing based on results in comparison to the total costs. However, since they stick to the MaxOccupancy of 90% the output of the box division model based on this value is used in the reallocation model.

5.2 Output of the Reallocation Model

This section provides the results of the reallocation model for Benchmark. Section 5.2.1 discusses the extraction of reallocations on removed carriers. Section 5.2.2 shows the execution of carrier-type changes. Section 5.2.3 provides the output of simulating the execution of OTM reallocation. Section 5.2.4 shows the results of the execution of the OTO reallocations.

5.2.1 Extraction of Reallocations on Removed Carriers

As shown in Figure 34, the first step of the reallocation model is the extraction of reallocations located on removed carrier types. Figure 45 shows the schedule for the extraction of reallocations that are placed on removed carrier types. For each order picker, the allocated jobs can be seen including the completion time and which shuttles they are located over time.

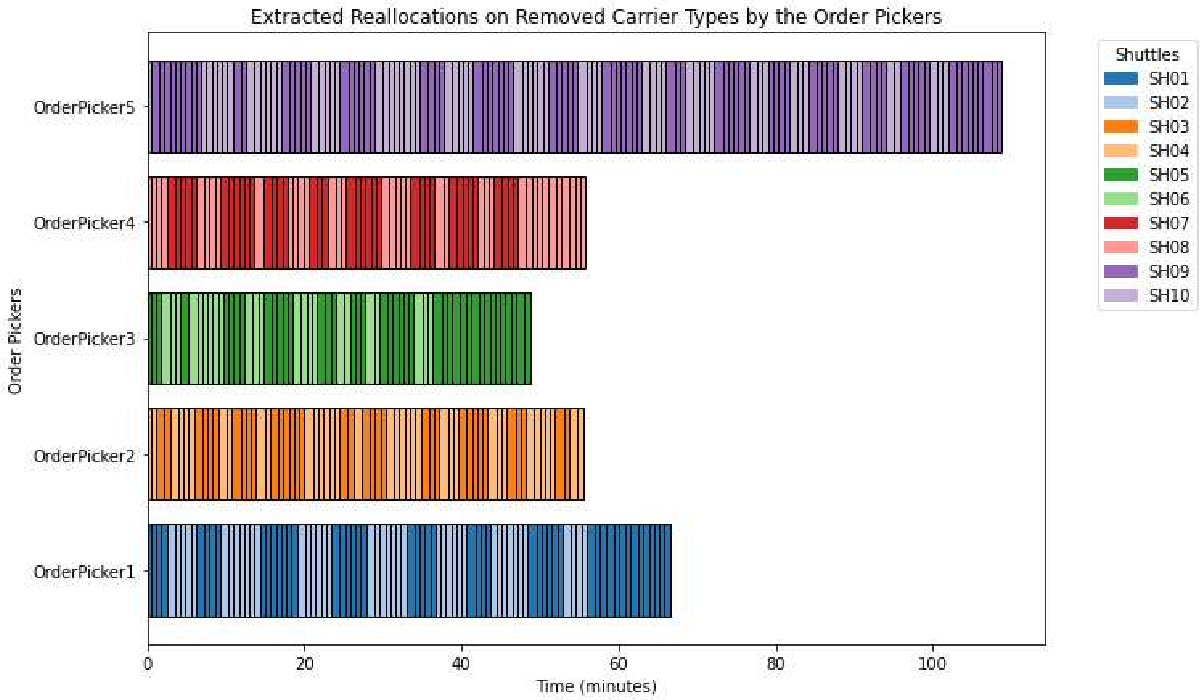

Figure 45: Overview of the Extracted Reallocations by the five order pickers

Next to a graphical overview of the jobs for the order picker, a summary of this step in the reallocations model is created as shown in Figure 46. The total completion time and total costs of the extracted reallocations can be seen for each order picker. As discussed, the internal rate of an order picker is 60 euros per hour or one euro per minute. Therefore, the total costs are determined by multiplying the internal rate of the order picker per minute by the total completion time for each order picker. In the end, the sum is taken of the cost per order picker leading to a completion time of approximately 335 minutes implying total costs of € 335.09.

```
Order Picker Summary for Extracted Reallocations:
 OrderPicker  NumberOfJobs  TotalCompletionTime (minutes) TotalCost
OrderPicker1            92                          66.50    €66.50
OrderPicker2            71                          55.42    €55.42
OrderPicker3            64                          48.67    €48.67
OrderPicker4            74                          55.75    €55.75
OrderPicker5           146                         108.75   €108.75

Total Summary for the Extracted Reallocations:
 TotalNumberOfJobs  TotalCompletionTime (minutes) TotalCost
               447                         335.09   €335.09
```

Figure 46: Summary of the first step of the reallocation model

5.2.2 Execution of Proposed Carrier Type Changes

The second step of the reallocation model executes the proposed carrier type changes by the box division model as shown in Figure 34. Figure 47 shows the carrier type changes that are performed by the order pickers.

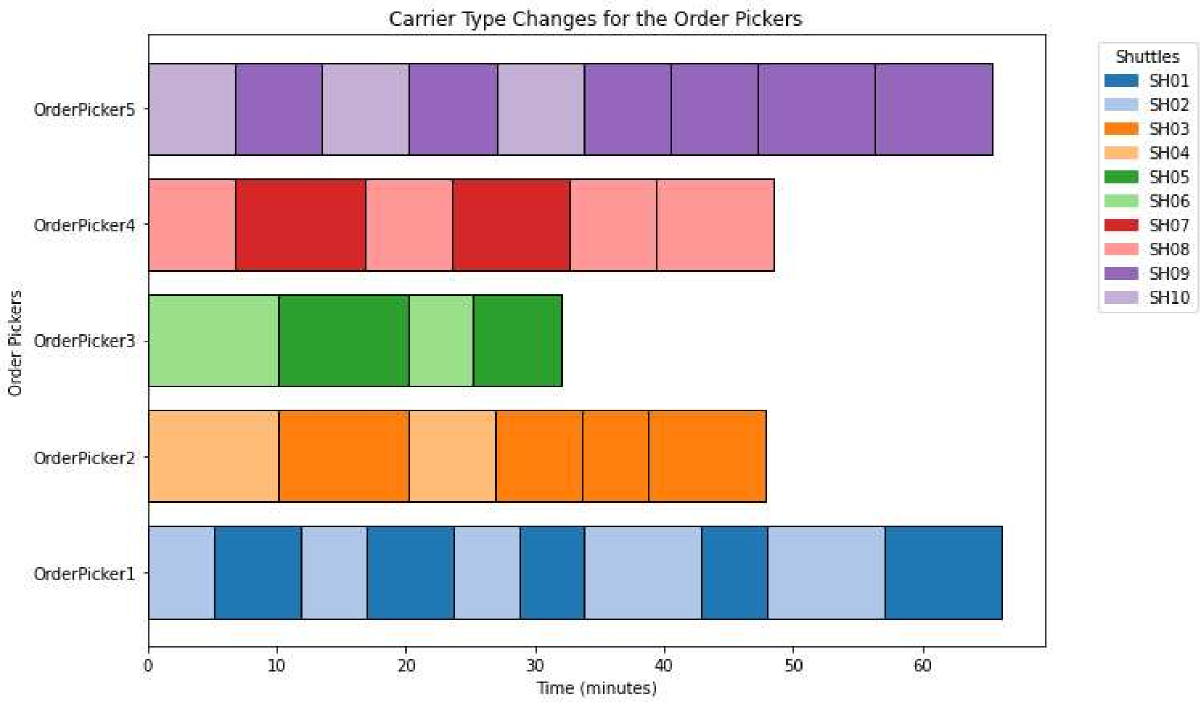

Figure 47: Overview of carrier type changes performed by the order pickers

Figure 48 shows the summary of the carrier type changes. There can be seen the completion time and total cost per order picker. Again, the internal cost rate of the order pickers has been used to calculate the total cost for each order picker. Eventually, the sum of the individual costs and time is calculated which leads to a total completion time of approximately 260 minutes and total costs of €259.92.

```
Carrier Type Changes Summary:
 OrderPicker  TotalCompletionTime (minutes) TotalCost
OrderPicker1                          66.17    €66.17
OrderPicker2                          47.83    €47.83
OrderPicker3                          32.00    €32.00
OrderPicker4                          48.50    €48.50
OrderPicker5                          65.42    €65.42

Total Summary for Carrier Type Changes:
 TotalCompletionTime (minutes) TotalCost
                       259.92   €259.92
```

Figure 48: Summary of the carrier changes output

5.2.3 Simulation of OTM Reallocations

The third step of the reallocation model simulates the execution of the OTM reallocations as shown in Figure 34. Section 4.3.3 explained that throughout this step the extracted OTM reallocations and OTM reallocations are performed. Therefore, this section is split up into the simulation of the extracted OTM reallocations and OTM reallocations.

Simulation of Extracted OTM Reallocations

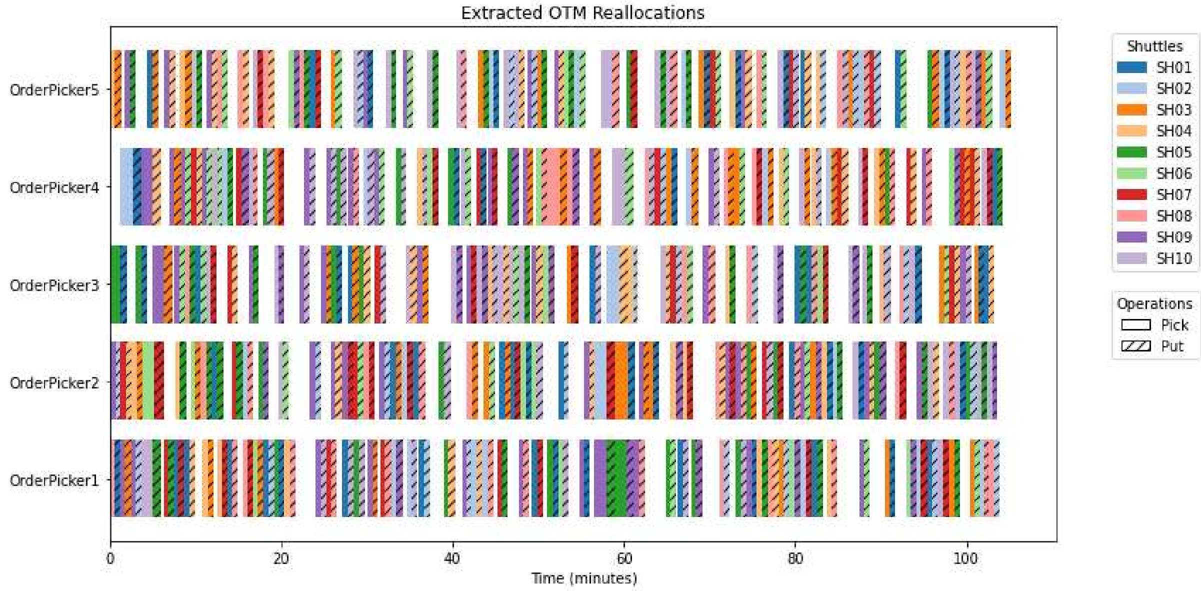

Figure 49: Example of a schedule for the execution of extracted OTM reallocations

Figure 49 shows an impression of a schedule created for the execution of the OTM reallocation. There can be seen that an OTM reallocation consists of pick and put actions. For each order picker, there can be seen which shuttles they are performing these actions over time for all assigned reallocations. Next to this schedule of reallocations per order pickers a summary of the extracted OTM reallocations is created as shown in Figure 50. In this summary, there can be seen the TotalCompletionTime and TotalCost per order picker. Furthermore, a summary of the total value of TotalCompletionTime and TotalCost is shown. The TotalCompletionTime was approximately 518 minutes which resulted in a TotalCost of € 518.88.

```
Extracted OTM Reallocations Order Picker Summary:
  OrderPicker1  TotalCompletionTime (minutes): 103.87    TotalCost: € 103.87  Number of Jobs: 60
  OrderPicker2  TotalCompletionTime (minutes): 103.60    TotalCost: € 103.60  Number of Jobs: 58
  OrderPicker3  TotalCompletionTime (minutes): 103.27    TotalCost: € 103.27  Number of Jobs: 51
  OrderPicker4  TotalCompletionTime (minutes): 102.90    TotalCost: € 102.90  Number of Jobs: 56
  OrderPicker5  TotalCompletionTime (minutes): 105.25    TotalCost: € 105.25  Number of Jobs: 56

Extracted OTM Reallocations Total Summary:
  TotalCompletionTime (minutes): 518.88  TotalCost: € 518.88
```

Figure 50: Summary on the extracted OTM reallocations

Simulation of OTM Reallocations

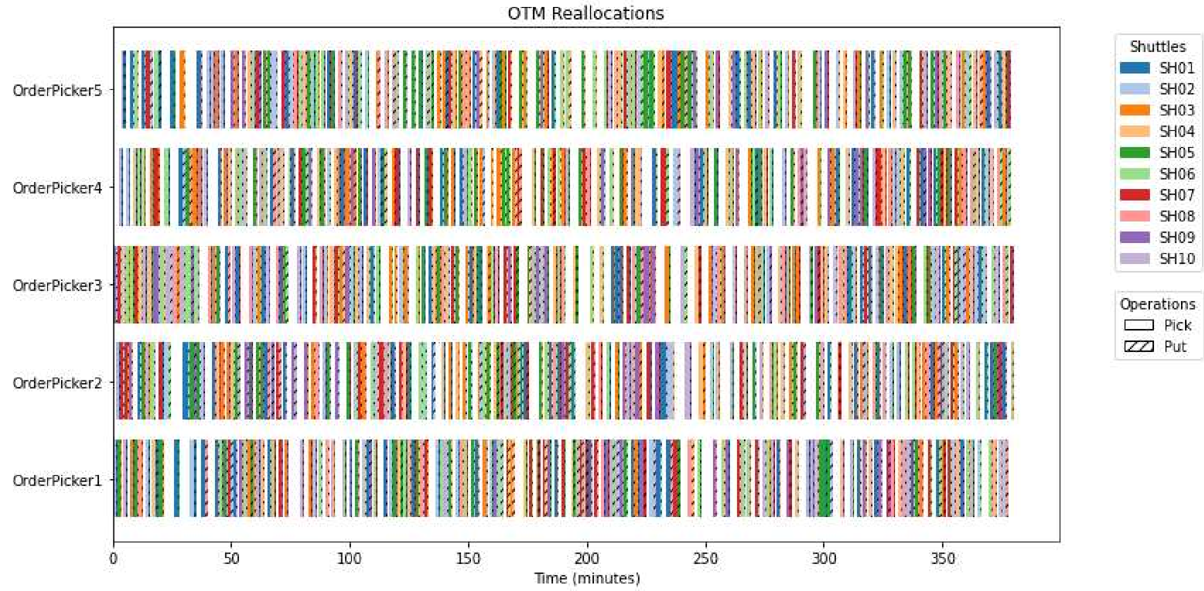

Figure 51: Example of a schedule for the execution of OTM reallocations

Another schedule is generated for the OTM Reallocations as can be seen in Figure 51. Due to the larger amount of reallocations, it immediately becomes clear that the execution of the OTM reallocations is more time-intensive. Again, a summary has been created to get an overview of the total time and costs of the OTM Reallocations as can be seen in Figure 52. For this schedule, the OTM Reallocations have a TotalCompletionTime of approximately 1890 minutes leading to TotalCost of € 1889.91.

```
OTM Reallocations Order Picker Summary:
  OrderPicker1  TotalCompletionTime (minutes): 377.74    TotalCost: € 377.74  Number of Jobs: 140
  OrderPicker2  TotalCompletionTime (minutes): 379.39    TotalCost: € 379.39  Number of Jobs: 136
  OrderPicker3  TotalCompletionTime (minutes): 380.41    TotalCost: € 380.41  Number of Jobs: 144
  OrderPicker4  TotalCompletionTime (minutes): 376.48    TotalCost: € 376.48  Number of Jobs: 135
  OrderPicker5  TotalCompletionTime (minutes): 375.89    TotalCost: € 375.89  Number of Jobs: 138

OTM Reallocations Total Summary:
  TotalCompletionTime (minutes): 1889.91 TotalCost: €1889.91
```

Figure 52: Summary of the OTM reallocations

As discussed in Section 4.3.3, the simulation of executing OTM reallocations is performed over 1000 iterations to provide a better impression of the total costs due to the randomness of the put location selection. After completing 1000 iterations a summary is provided on the average time and costs as shown in Figure 53. The average total costs for extracted OTM reallocations are € 497.84 while the OTM reallocations cost € 1880.42 on average. In total, the average total cost for performing all OTM reallocations is determined at € 2378.26

```
Average Total Cost for Extracted OTM Reallocations over 1000 iterations: €497.84
Average Total Cost for OTM Reallocations over 1000 iterations: €1880.42

Average Total Cost for all OTM Reallocations over 1000 iterations: €2378.26
```

Figure 53: Total Summary of executing all OTM Reallocations

5.2.4 Scheduling OTO Reallocations using Heuristics

The fourth step of the reallocation model schedules the OTO reallocations as shown in Figure 34. As discussed in Section 4.3.4, the extracted OTO reallocations and OTO reallocations are performed in this step of the reallocation model. Therefore, this section is split up into scheduling extracted OTO reallocations and OTO reallocations.

Scheduling Extracted OTO Reallocations

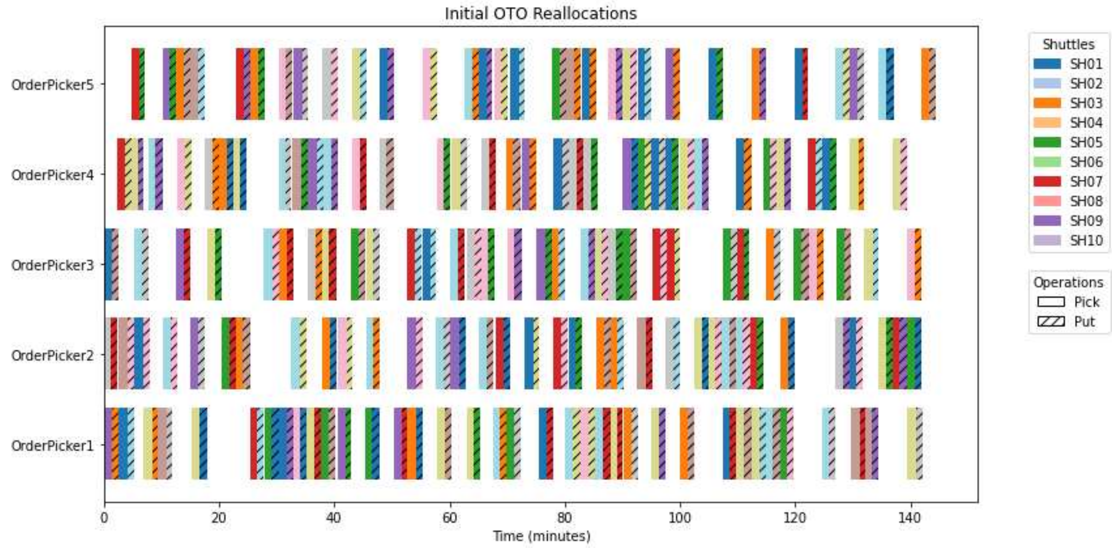

Figure 54: Schedule for extracted OTO Reallocations using only the nearest neighbour heuristic

Figure 54 shows the schedule generated by using only the nearest neighbour algorithm as a constructive heuristic. For each order picker, there can be seen on which shuttle they are working over time. For each reallocation, a pick and put location are shown in the schedule.

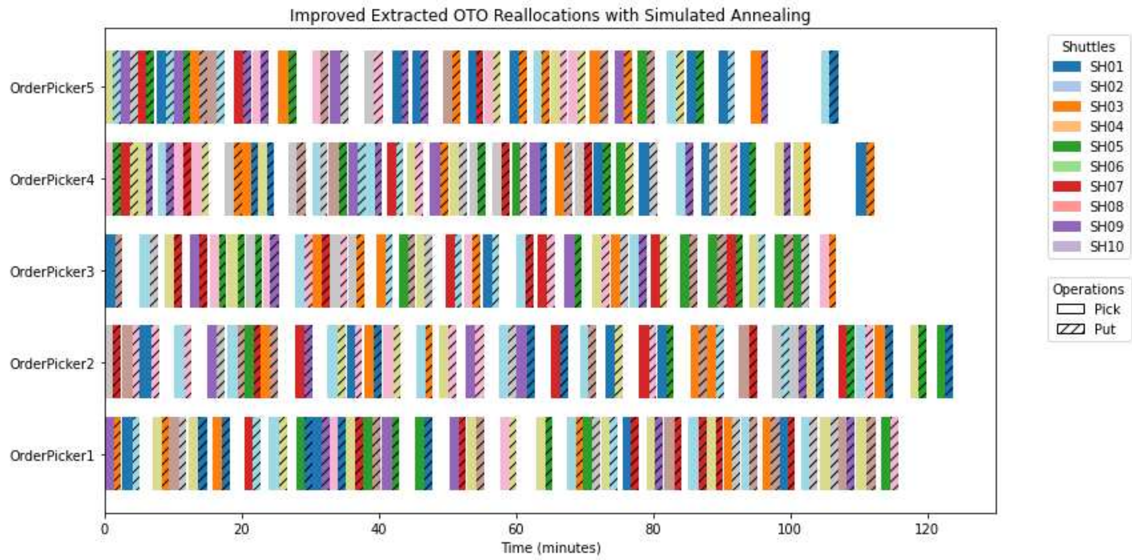

Figure 55: Extracted OTO reallocations using nearest neighbour and simulated annealing

Next to solely using the constructive heuristic, Figure 55 shows the use of simulated annealing in combination with the nearest neighbour heuristic. It can be seen that the completion time until finishing the last job for all order pickers was reduced for several order pickers. Compared to Figure 54, the completion time decreased from over 140 minutes to about 120 minutes or less for all order pickers. This experiment used the simulated annealing settings as shown in Figure 41.

```
Extracted OTO Reallocations Order Picker Summary:

Initial Total Cost: €709.67
Improved Total Cost: €565.67
Cost Reduction: €144.00

Simulated Annealing successfully reduced the total cost.
```

Figure 56: Summary of extracted OTO reallocations using nearest neighbour and simulated annealing

Next to the plots, a summary containing an overview of all costs is created as shown in Figure 56. The summary contains both the initial total costs using the constructive heuristic solely and improved total costs for the improvement heuristics. Also, the cost reduction that the improvement heuristic managed to obtain is calculated.

Table 15: Total cost using constructive heuristics for extracted OTO reallocations

Constructive Heuristic	Initial Total Costs
Nearest Neighbour	€ 709.67
Farthest Neighbour	€ 681.82

The total costs for executing the extracted OTO reallocations using only the nearest neighbour or farthest neighbour heuristic are shown in Table 15. To improve the solutions of the constructive heuristics there have been performed several experiments using the improvement heuristic. These experiments use different combinations for the initial temperature, iterations, and alpha as shown in Table 16.

Table 16: Cost of using constructive heuristics with improvement heuristic for extracted OTO reallocations

Constructive Heuristic	Initial Temp	Iterations	Alpha	Improved Costs Simulated Annealing
Nearest Neighbour	200	50	0.99	€ 565.67
Nearest Neighbour	300	100	0.99	€ 565.67
Nearest Neighbour	400	200	0.99	€ 565.67
Nearest Neighbour	1000	500	0.99	€ 565.67
Farthest Neighbour	200	50	0.99	€ 563.15
Farthest Neighbour	300	100	0.99	€ 563.15
Farthest Neighbour	400	200	0.99	€ 563.15
Farthest Neighbour	1000	500	0.99	€ 563.15

These experiments show that the simulated annealing algorithm managed to decrease the improved costs for all experiments. It is remarkable that for both constructive heuristics the improvement heuristic did only manage to get one value for the improved cost. The best solution could be found in the experiment using the farthest neighbour heuristic resulting in an improved total cost of € 563.15 where the schedule of this solution is shown in Figure 57. The limited variety in improved cost can be explained because there are limited options for moving reallocations between order pickers. This is because many shuttles are already fully occupied by order pickers over time making it hard to create feasible neighbour solutions since a shuttle cannot be occupied by two order pickers at the same time. All in all, the use of simulated annealing as an improvement heuristic resulted in a significant cost reduction compared to the initial solution of the farthest neighbour heuristic delivering a better schedule.

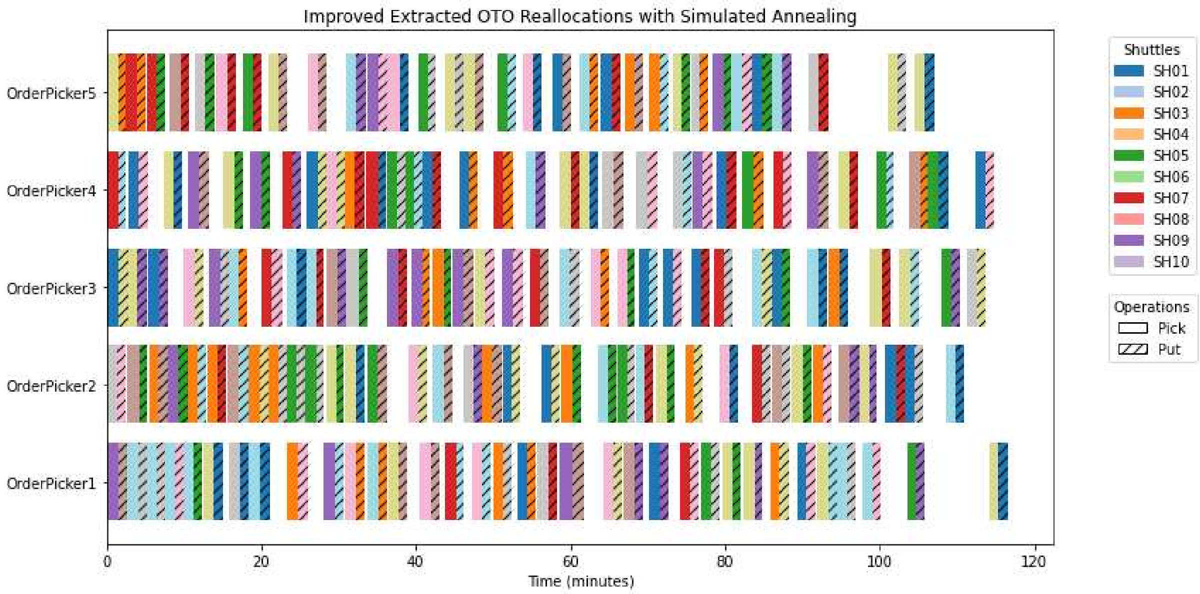

Figure 57: Best schedule for the extracted OTO reallocations

Scheduling OTO Reallocations

Similar to the extracted OTO reallocations schedules have been created for all OTO Reallocations using the nearest neighbour and farthest neighbour constructive heuristics. Figure 58 shows an example of the schedule constructed using only the nearest neighbour algorithm.

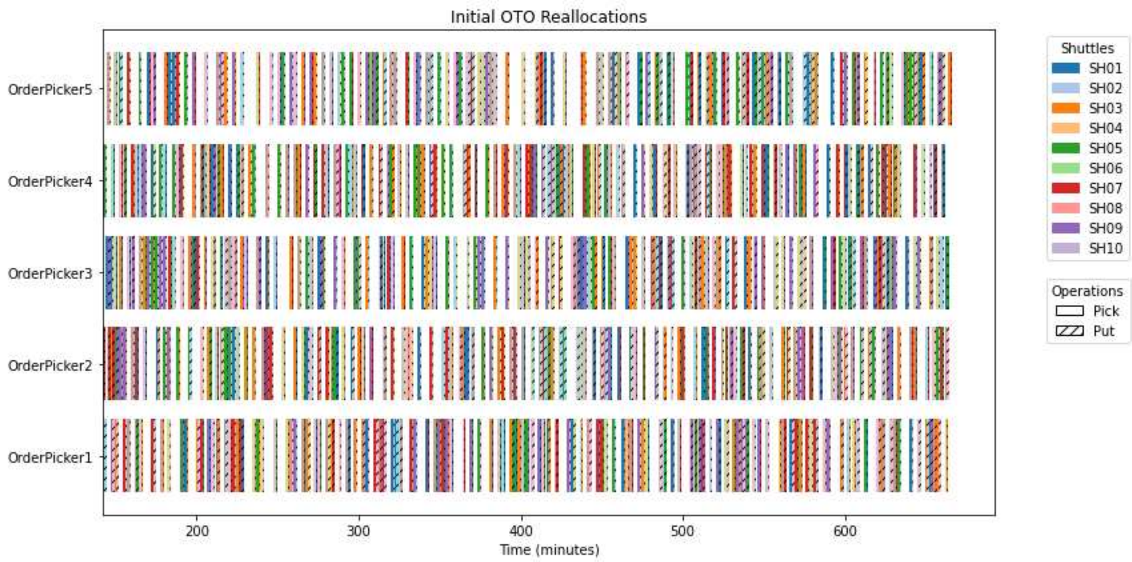

Figure 58: Schedule created using Nearest Neighbour for the OTO reallocations

Again, a summary has been created for the total cost using the chosen constructive and improvement heuristic depending on input settings as shown in Figure 59.

```
OTO Reallocations Order Picker Summary:

Initial Total Cost: €3321.17
Improved Total Cost: €2095.27
Cost Reduction: €1225.90

Simulated Annealing successfully reduced the total cost.
```

Figure 59: Total Summary for OTO reallocations

Table 17 shows an overview of the initial solution created by solely the nearest neighbour or farthest neighbour heuristics.

Table 17: Cost of constructive heuristics for OTO Reallocations

Constructive Heuristic	Initial Total Costs
Nearest Neighbour	€ 3321.17
Farthest Neighbour	€ 3217.33

In addition to running the constructive heuristic solely several experiments have again been performed to see which combination performs better as shown in Table 18.

Table 18: Cost by using constructive heuristics with improvement heuristic for the OTO Reallocations

Constructive Heuristic	Initial Temp	Iterations	Alpha	Improved Costs Simulated Annealing
Nearest Neighbour	200	50	0.99	€ 2104.28
Nearest Neighbour	300	100	0.99	€ 2097.50
Nearest Neighbour	400	200	0.99	€ 2095.27
Nearest Neighbour	1000	500	0.99	€ 2088.60
Farthest Neighbour	200	50	0.99	€ 2160.00
Farthest Neighbour	300	100	0.99	€ 2126.72
Farthest Neighbour	400	200	0.99	€ 2104.27
Farthest Neighbour	1000	500	0.99	€ 2099.12

Contrary to the results of the experiments using simulated annealing at the extracted OTO reallocations, the results for the experiments at the OTO reallocations show more difference in improved total costs. This can be explained by the fact that there are far more OTO reallocations than extracted OTO that must be performed which creates more possibilities for finding feasible neighbour solutions. From the experiments, it can be concluded that the nearest neighbour heuristic performed the best with an initial temperature of 1000 and 500 iterations. The improved costs for the OTO reallocations turned out to be € 2088.60. The schedule that comes out of this experiment is shown in Figure 60.

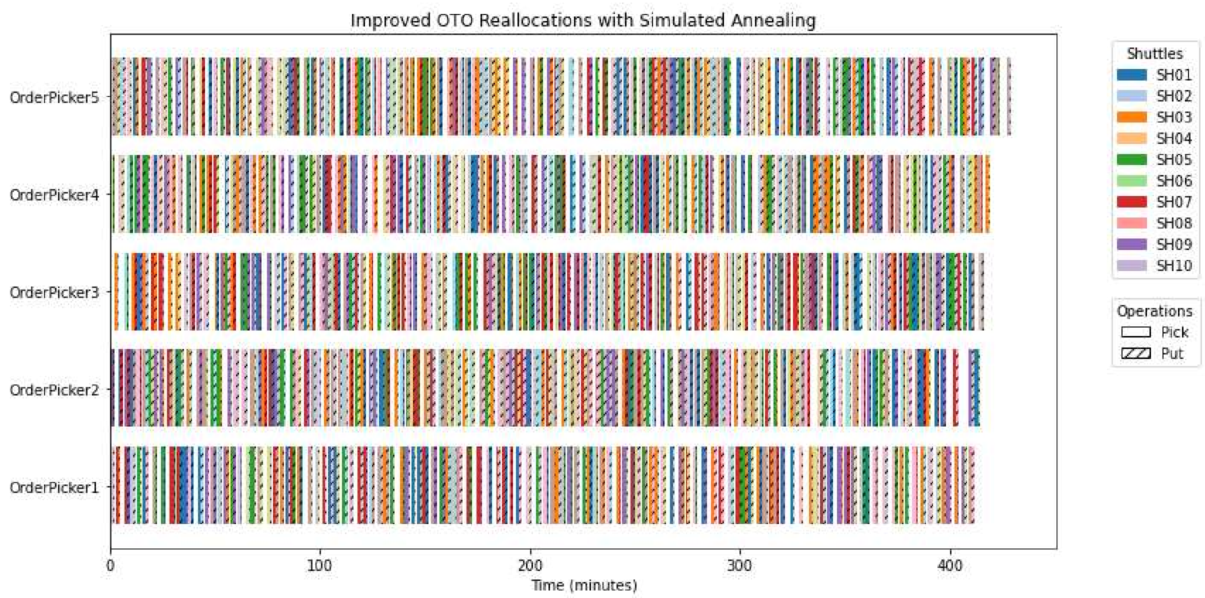

Figure 60: Best schedule for the OTO Reallocations

5.3 Cost Analysis

After calculating both material costs using the box division model and labour costs using the reallocation model there is performed a cost analysis on implementing the box allocation method. Table 19 shows an overview of all costs surrounding the implementation of the box allocation method.

Table 19: Overview of Total Costs of the Box Allocation Method

Operation	Cost
Material Costs	€ 2131.46
Extracting Reallocations	€ 335.09
Carrier Type Changes	€ 259.92
Execute Extracted OTM Reallocations	€ 497.84
Execute Extracted OTO Reallocations	€ 563.15
OTM Reallocations	€ 1880.42
OTO Reallocations	€ 2088.60
Total Cost	€ 7756.48

The total costs are determined at the amount of € 7756.48 consisting of both the material and labour costs. Considering the fact that an additional Kardex shuttle would cost approximately €75,000, the investment in the box allocation method is far lower than in an additional Kardex shuttle. Also, since it will be extremely hard to add more Kardex shuttles in the current ZKDX1001 zone using the box allocation method would be a good option for Benchmark. This would allow Benchmark to get more storage space without the need to invest in additional space and Kardex shuttles.

5.4 Conclusion

The results of the box division model and reallocation model showed that Benchmark could get a significant increase in storage efficiency for the Kardex shuttles by implementing the box allocation method. The box division model calculated that the unoccupied volume increased from 38.9 m3 in the current situation to 51.5 m3 while maintaining a maximum occupancy rate of 90%. Moreover, Benchmark could decrease the maximum occupancy rate up to 85% to acquire an unoccupied volume of 56.4 m3. This unoccupied volume can be used for the placement of additional carriers and boxes of a certain carrier type as shown in Table 8. By doing this the storage capacity for each box type in the Kardex shuttles could increase even more. The cost analysis showed that by using a maximum occupancy of 90% the box allocation method could be implemented at a cost of € 7756.48.

Chapter 6 – Conclusion and Recommendations

This chapter formulates a conclusion on this research. Furthermore, several limitations are discussed and recommendations are made to Benchmark. Finally, several topics for future research are provided.

6.1 Conclusion

Below can be seen the main research question formulated for this research:

" How can more storage efficiency be obtained by using an optimized box allocation method for the Kardex shuttles at Benchmark? "

This research developed a box allocation method consisting of the box division model and reallocation model.

Box Division Model

The box division model is developed to determine the recommended box type for supplied SKUs and propose reallocations for boxes stored in the Kardex shuttles. The box division model uses three different reallocation methods to change the box division in the Kardex shuttles.

- OTO Reallocations: The box division model proposes several One-to-One (OTO) reallocations that lead to instant volume savings in the Kardex shuttles. It changes a large box type by one box of a smaller box type to achieve volume savings in the Kardex shuttles.
- OTM Reallocations: One-to-Many (OTM) reallocations are proposed by the box division model if volume can be saved by changing one box to multiple other boxes of a different box type for boxes stored in the Kardex shuttles. This reallocation method differs from OTO reallocations since multiple boxes are involved for the recommended box type in the OTM reallocation.
- Carrier Type Changes: Another reallocation method proposed by the box division model is carrier type changes. Carrier type changes change the carrier types of carriers, resulting in the placement of boxes of a different box type on carriers selected for carrier type changes. This is useful if a high occupancy rate for a certain box type in the Kardex shuttles is faced. The box division model uses a desired maximum occupancy rate per box type and proposes carrier type changes to achieve this desired maximum occupancy rate.

The results of the box division model clearly showed that the use of the B004 carrier type is not desired. It could be observed that carrier type changes were only performed on B004 carriers which were changed to another carrier type. Also, supplied SKUs tend to prefer smaller box types instead of the B004 box type. Using the reallocations provided by the box division model, Benchmark can achieve an unoccupied volume of 51.5 m3 which is an improvement over the current 38.9 m3. This unoccupied volume can be used for the placement of additional carriers according to Table 8. The material costs of these reallocations are determined at € 2131.46.

Reallocation Model

The reallocation model is developed to create insights into the time and labour costs of executing the reallocations proposed by the box division model. This is because the box division model only provides insights into proposed reallocations and their resulting total material costs reallocations. The reallocation model provides insights into the execution of the reallocations over several steps and provides a calculation of the estimated completion time and labour costs for each step. The total labour costs of executing the reallocations were determined at € 5625.02.

All in all, there has been a significant improvement of 12.6 m3 in the unoccupied volume of the Kardex shuttles which can be realized at the total cost of € 7756.48.

6.2 Recommendations

Benchmark should implement the box allocation method to decrease the occupied volume in the Kardex shuttles. They could test the real-world usability in the warehouse and see if the box allocation method works as intended. After implementing the box allocation method it is recommended to use the box division model and reallocation model again in case the occupancy rate of the box types in the Kardex shuttles becomes too high in the future. Benchmark could then decide whether it is worth performing the reallocations based on the total costs and created savings in the Kardex shuttles.

Another recommendation is to perform adjustments to the carrier heights. By executing the reallocations from the reallocation methods, an increase in unoccupied volume is created. This unoccupied volume can be used for the placement of additional carriers after the carrier height adjustments are performed. This would decrease the occupancy rate for the box type in the Kardex shuttles even more in the future.

Finally, the employees should be trained on using the box allocation method and applications developed for it. Since the applications could be complex to understand for employees, it is important to show employees the value of using them at Benchmark. After employees are trained in using the applications, the usability becomes more straightforward for them.

6.3 Limitations

This research aimed to improve the storage efficiency by implanting a box allocation method at the original ten Kardex shuttles placed at the ZKDX1001 zone. Benchmark also has other zones containing Kardex shuttles and two recently acquired two Kardex shuttles. It could be valuable for Benchmark to investigate the possible savings in these Kardex shuttles as well. Next to the other Kardex shuttles, Benchmark could also try to develop similar models for the pallet area which is present in the ZKDX1001 zone. These models could be adapted by the logic coming from the box allocation method.

6.4 Future Research

As discussed in the problem definition the FIFO policy could be considered as another core problem for the inefficient use of storage capacity in the Kardex shuttles. Benchmark mentioned that they do not want to change the FIFO policy, but this could have a significant impact on the unoccupied volume of the Kardex shuttles. Therefore, Benchmark could consider investigating the impact of dropping the FIFO policy on the storage efficiency in the Kardex shuttles.

Another topic for future research could be to fully integrate the recommended boxes for supplied SKUs in the software systems. If the box allocation method will be used every month, it could make sense to provide the recommended box type for supplied SKUs via a central place to the employees who are selecting the box types at the incoming goods department. Implementing the recommended box type in a software system could directly provide the recommended box types to these employees on a monthly basis after running the models.

References

Agarwal, P. K., Matoušek, J., & Suri, S. (1992). Farthest neighbors, maximum spanning trees and related problems in higher dimensions. Computational Geometry, 1(4), 189–201. https://doi.org/10.1016/0925-7721(92)90001-9

Ardjmand, E., Shakeri, H., Singh, M., & Bajgiran, O. S. (2018). Minimizing order picking makespan with multiple pickers in a wave picking warehouse. *International Journal of Production Economics*, *206*, 169–183. https://doi.org/10.1016/j.ijpe.2018.10.001

Atmaca, E., & Ozturk, A. (2013). Defining order picking policy: A storage assignment model and a simulated annealing solution in AS/RS systems. Applied Mathematical Modelling, 37(7), 5069–5079. https://doi.org/10.1016/j.apm.2012.09.057

Battini, D., Calzavara, M., Persona, A., & Sgarbossa, F. (2016). Dual-tray Vertical Lift Modules for Fast Order Picking. 14TH IMHRC PROCEEDINGS. https://og.mhi.org/downloads/learning/cicmhe/colloquium/2016/04-Battini_Calzavara_Persona_Sgarbossa_paper.pdf

Çelk, M., & Süral, H. (2013). Order picking under random and turnover-based storage policies in fishbone aisle warehouses. Iie Transactions, 46(3), 283–300. https://doi.org/10.1080/0740817x.2013.768871

Đukić, G., Opetuk, T., & Lerher, T. (2015). A throughput model for a dual-tray Vertical Lift Module with a human order-picker. International Journal of Production Economics, 170, 874–881. https://doi.org/10.1016/j.ijpe.2015.04.009

Gil-Borrás, S., Pardo, E. G., Alonso-Ayuso, A., & Duarte, A. (2021). A heuristic approach for the online order batching problem with multiple pickers. Computers & Industrial Engineering, 160, 107517. https://doi.org/10.1016/j.cie.2021.107517

Henn, S., & Wäscher, G. (2012). Tabu search heuristics for the order batching problem in manual order picking systems. European Journal of Operational Research, 222(3), 484–494. https://doi.org/10.1016/j.ejor.2012.05.049

Horselenberg, J. (2023). Improving warehouse efficiency: A study of the KPIs and picking process at Benchmark - University of Twente Student Theses. https://purl.utwente.nl/essays/95267

Jansman, A. J. (2014). Een onderzoek naar het voorbereiden van de implementatie van Kardex VLM systemen in het magazijn van Benchmark Electronics - University of Twente Student Theses. https://purl.utwente.nl/essays/65563

Lee, M., & Elsayed, E. A. (2005). Optimization of warehouse storage capacity under a dedicated storage policy. International Journal of Production Research, 43(9), 1785–1805. https://doi.org/10.1080/13528160412331326496

Lenoble, N., Frein, Y., & Hammami, R. (2018). Order batching in an automated warehouse with several vertical lift modules: Optimization and experiments with real data. European Journal of Operational Research, 267(3), 958–976. https://doi.org/10.1016/j.ejor.2017.12.037

Lerher, T. (2017). Design of Experiments for Identifying the Throughput Performance of Shuttle-Based Storage and Retrieval Systems. Procedia Engineering, 187, 324–334. https://doi.org/10.1016/j.proeng.2017.04.382

Menéndez, B., Pardo, E. G., Alonso-Ayuso, A., Molina, E., & Duarte, A. (2017). Variable Neighborhood Search strategies for the Order Batching Problem. Computers & Operations Research, 78, 500–512. https://doi.org/10.1016/j.cor.2016.01.020

Park, B., & Lee, M. (2007). Closest open location rule under stochastic demand. International Journal of Production Research, 45(7), 1695–1705. https://doi.org/10.1080/00207540600855007

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45–77. https://doi.org/10.2753/mis0742-1222240302

Petersen, C. G., Aase, G. R., & Heiser, D. R. (2004). Improving order-picking performance through the implementation of class-based storage. International Journal of Physical Distribution & Logistics Management, 34(7), 534–544. https://doi.org/10.1108/09600030410552230

Rahman, M. A., & Parvez, H. (2021). Repetitive nearest neighbor based simulated annealing search optimization algorithm for traveling salesman problem. OAlib, 08(06), 1–17. https://doi.org/10.4236/oalib.1107520

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

Tenhagen, C. (2018). Increasing space efficiency within a Vertical Lift Module at Benchmark Electronics, Inc. https://essay.utwente.nl/76737/1/Thesis%20Tenhagen.pdf

Toth, P. (2000). Optimization engineering techniques for the exact solution of NP-hard combinatorial optimization problems. European Journal of Operational Research, 125(2), 222–238. https://doi.org/10.1016/s0377-2217(99)00453-1

Yu, Y., & De Koster, R. (2013). On the suboptimality of full turnover-based storage. International Journal of Production Research, 51(6), 1635–1647. https://doi.org/10.1080/00207543.2011.654012

Appendix A – Location List Report from PPG

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Location	X-step	Y-step	Material	Current quantity	Dedicated quantity	Handling Unit	Put date	Lot	Qualification	Storage unit	Carrier	Shelf	Bin
4	SH06-008-003-03	3	3	ASM4022_438_75306-LF	128	0		2023-03-31 1	MBA467851	A0000N	SH06	008	01	B002
11	SH04-006-009-01	9	1	ASM4022_711_85891-LF	8	0	B002-0002	2023-12-15 1	ANP005162-2350-00752	A0000N	SH04	006	01	B002
12	SH02-001-020-01	20	1	ASM4022_438_09089-LF	1000	0	B001-3153	2024-02-05 1	ANP002699-2406-00234	A0000N	SH02	001	01	B001
15	SH06-018-003-01	3	1	ASM4022_685_40902-LF	10	0	B003-1361	2024-02-20 1	ANP000499-2408-00247	A0000N	SH06	018	01	B003
24	SH05-042-004-02	4	2								SH05	042	01	B001
28	SH02-046-013-02	13	2	ASM4022_693_80701-LF	3	0		2023-03-31 1	MBA409100	A0000N	SH02	046	01	B001
31	SH06-012-007-03	7	3								SH06	012	01	B002
33	SH05-009-005-03	5	3	ASM4022_668_81621-LF	30	0	B002-1181	2024-03-21 1	ANP009391-2412-00756	A0000N	SH05	009	01	B002
34	SH03-008-009-04	9	4	ASM4022_438_34209-LF	73	0	B002-2057	2024-03-04 1	ANP008771-2410-00099	A0000N	SH03	008	01	B002
41	SH08-038-005-01	5	1	ASM4022_472_56922-LF	15	0	B004-0663	2024-03-05 0	ANP005851-2410-00231	A0000N	SH08	038	01	B004
53	SH08-048-004-02	4	2								SH08	048	01	B001
54	SH08-047-020-02	20	2								SH08	047	01	B001
57	SH07-030-004-01	4	1	ASM4022_711_21431-LF	8	0	B003-3132	2024-04-08 1	ANP009124-2413-00276	A0000N	SH07	030	01	B003
74	SH07-011-002-04	2	4				B001-0959				SH07	011	01	B001
75	SH04-021-002-01	2	1	ASM4022_685_71132-LF	15	0	B003-1911	2024-04-23 0	ANP007412-2417-00158	A0000N	SH04	021	01	B003
77	SH08-027-001-02	1	2				B004-1807				SH08	027	01	B004
79	SH05-040-010-02	10	2	ASM4022_476_23022-LF	8	0	B003-1489	2024-02-27 0	ANP008401-2409-00210	A0000N	SH05	040	01	B003
84	SH09-002-001-01	1	1				B001-1896				SH09	002	01	B001
88	SH08-014-010-01	10	1	ASM4022_476_01401-LF	6	0	B003-2600	2024-03-22 1	ANp004590-2412-00778	A0000N	SH08	014	01	B003
92	SH01-009-08-01	8	1	ASM4022_660_28752-LF	1	0		2023-03-31 1	MBA455323	A0000N	SH01	009	01	B002
95	SH04-007-003-01	3	1	ASM4022_698_29811-LF	8	0	B002-3023	2024-04-10 1	ANP005468-2415-00311	A0000N	SH04	007	01	B002
99	SH09-042-010-02	10	2								SH09	042	01	B001
01	SH02-004-010-03	10	3	ASM4022_472_65202-LF	10	0	B002-3728	2023-06-22 1	ANZ003159-2325-00281	A0000N	SH02	004	01	B002
12	SH03-045-015-02	15	2								SH03	045	01	B001
13	SH07-011-003-03	3	3	ASM4022_438_41288-LF	1	0		2023-03-31 1	MBA380789	A0000N	SH07	011	01	B001
17	SH06-005-007-03	7	3	ASM4022_438_40734-LF	11	0	B002-0162	2024-02-15 1	ANP006917-2407-00707	A0000N	SH06	005	01	B002
20	SH08-007-009-02	9	2	ASM4022_438_33382-LF	53	0		2023-03-31 1	MBA435340	A0000N	SH08	007	01	B001
40	SH07-002-003-01	3	1	ASM4022_478_00740-LF	4	0	B001-1028	2024-04-15 1	ANP000983-2416-00005	A0000N	SH07	002	01	B001
47	SH08-006-017-01	17	1				B001-4754				SH08	006	01	B001
60	SH06-003-005-04	5	4				B001-1433				SH06	003	01	B001
63	SH09-030-034-01	34	1	ASM4022_693_97832-LF	4	0	B006-0141	2024-04-03 1	ANP007462-2414-00312	A0000N	SH09	030	01	B006
66	SH01-003-13-04	13	4				B001-4285				SH01	003	01	B001
76	SH04-040-039-01	39	1								SH04	040	01	B006
78	SH05-003-012-03	12	3	ASM4022_439_91043-LF	94	0		2023-03-31 1	MBA391807	A0000N	SH05	003	01	B001
82	SH05-017-004-02	4	2	ASM4022_438_50506-LF	3	0	B003-1651	2023-11-09 0	ANP001163-2345-00289	A0000N	SH05	017	01	B003
86	SH05-046-002-04	2	4								SH05	046	01	B001
87	SH07-044-013-04	13	4								SH07	044	01	B001
95	SH09-032-002-02	2	2	ASM4022_685_40962-LF	13	0	B004-0112	2024-01-17 0	ANP001317-2403-00418	A0000N	SH09	032	01	B004
97	SH10-009-007-02	7	2	ASM4022_478_00553-LF	14	0	B003-1217	2024-03-20 1	ANP005151-2412-00487	A0000N	SH10	009	01	B003
01	SH02-039-021-01	21	1	ASM4022_704_21961-LF	1	0	B006-0383	2024-04-09 1	ANP000059-2329-00247	A0000N	SH02	039	01	B006
05	SH08-002-017-03	17	3	ASM4022_438_42642-LF	37	0	B001-2222	2023-04-12 1	ANZ004807-2314-00504	A0000N	SH08	002	01	B001
06	SH03-049-002-02	2	2	ASM4022_664_18192-LF	4	0	B003-1421	2023-11-30 1	ANP005809-2348-00236	A0000N	SH03	049	01	B003
07	SH02-017-010-02	10	2	ASM4022_636_19303-LF	4	0	B003-2898	2024-03-22 1	ANP005193-2412-00998	A0000N	SH02	017	01	B003
08	SH03-011-006-02	6	2	ASM4022_438_34865-LF	34	0	B001-4956	2023-10-13 1	ANP005118-2340-01353	A0000N	SH03	011	01	B001
10	SH08-012-024-01	24	1	ASM4022_682_00113-LF	1	0	B006-0346	2024-04-16 1	MR0000417	A0000N	SH08	012	01	B006
12	SH09-009-003-01	3	1	ASM4022_640_72331-LF	9	2	B003-1589	2024-03-20 1	ANP009465-2412-00339	A0000N	SH09	009	01	B003
13	SH04-013-007-02	7	2	ASM4022_694_24132-LF	6	0	B003-0679	2024-03-18 1	ANP005847-2412-00098	A0000N	SH04	013	01	B003
14	SH06-047-014-01	14	1								SH06	047	01	B001
19	SH02-003-006-03	6	3				B001-2552				SH02	003	01	B001
20	SH01-006-07-01	7	1	ASM4022_438_42581-LF	4	0	B002-2153	2023-11-13 0	ANP000403-2345-00917	A0000N	SH01	006	01	B002
33	SH03-004-003-03	3	3	ASM4022_693_28741-LF	2	0		2023-03-31 1	MBA392512	A0000N	SH03	004	01	B002
34	SH07-039-008-02	8	2	ASM0051_107_00035-LF	6	0		2023-03-31 1	MBA426343	A0000N	SH07	039	01	B001

Figure 61: Screenshot of the Location list coming from PPG

Appendix B – Overview of a Carrier Used in Kardex Shuttles

Tray type	75 3 Medium 1 0 0
Quantity per unit	102 pcs.
Max. load per tray	390 kg
Clear tray width	3650 mm
Clear tray depth	864 mm
Storage area	3,154 m²
Max. height of stored goods	727 mm

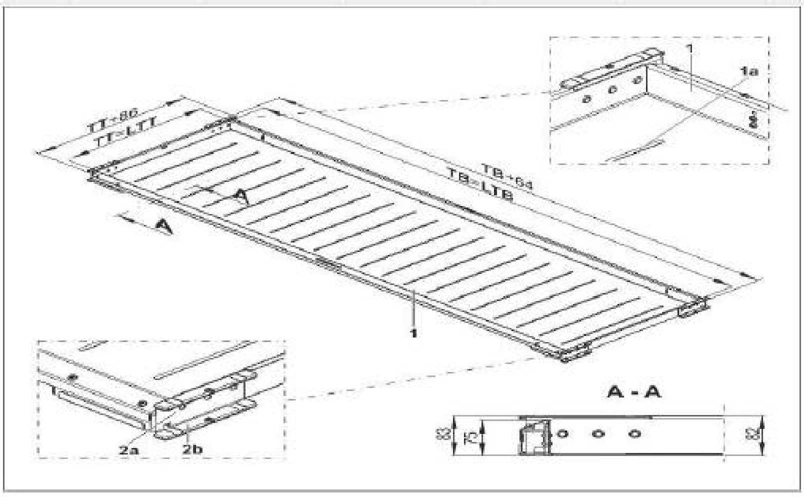

Figure 62: Specifications of a Carrier in use by Benchmark

Appendix C – Fixed Location List from LN

	D	E	F	G	H	I	
1	Company	Warehouse		Location		Item	
2	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_102_00002-LF
3	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00111-LF
4	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00121-LF
5	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00131-LF
6	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00141-LF
7	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00151-LF
8	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00161-LF
9	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00171-LF
10	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00181-LF
11	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00191-LF
12	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00201-LF
13	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00231-LF
14	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00241-LF
15	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00251-LF
16	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00271-LF
17	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00301-LF
18	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00321-LF
19	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00331-LF
20	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00341-LF
21	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00371-LF
22	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00411-LF
23	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00421-LF
24	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00431-LF
25	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00441-LF
26	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00451-LF
27	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00471-LF
28	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_103_00591-LF
29	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_107_00026-LF
30	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_107_00027-LF
31	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_107_00028-LF
32	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_107_00035-LF
33	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_107_00085-LF
34	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_107_00088-LF
35	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_107_00092-LF
36	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_108_00051-LF
37	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_110_00021-LF
38	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_110_00031-LF
39	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_240_00571-P
40	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_240_00691-LF
41	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_240_00721-LF
42	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_240_01131-LF
43	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_240_01301-P
44	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_240_01431-P
45	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_240_01461-P
46	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_240_01471-P
47	1145	A0000N	MAIN STOCK WAREHOUSE-AN	ZKDX1001	ZKDX1001		ASM0051_240_01501-P

Figure 63: Screenshot of the Fixed location list coming from LN

Appendix D – Time-Phased Planning from Rapid Response

	A	B	C	D	E	F	G	H	I	J
1	Prod									
2	Line	Site	Part	Description	Due Date	Recommended	Supply	Demand	Balance	Peg Part
25184			ASM0051_240_00831-LF	K21 BRACKET ASSEMBLY	05-22-24	05-22-24		1	-1	ASM4022_636_85619-LF
25185				K21 BRACKET ASSEMBLY	05-24-24	05-22-24	1		0	
25186				K21 BRACKET ASSEMBLY	02-14-25	02-17-25	6		6	
25187				K21 BRACKET ASSEMBLY	02-17-25	02-17-25		1	5	ASM4022_636_85619-LF
25188				K21 BRACKET ASSEMBLY	02-17-25	02-17-25		1	4	ASM4022_636_85619-LF
25189				K21 BRACKET ASSEMBLY	02-17-25	02-17-25		1	3	ASM4022_636_85619-LF
25190				K21 BRACKET ASSEMBLY	02-17-25	02-17-25		1	2	ASM4022_636_85619-LF
25191				K21 BRACKET ASSEMBLY	02-17-25	02-17-25		1	1	ASM4022_636_85619-LF
25192				K21 BRACKET ASSEMBLY	02-17-25	02-17-25		1	0	ASM4022_636_85619-LF
25193			**Sum**				**7**	**7**		
25194			ASM0051_240_00841-LF	REAR RAIL ASSY	05-22-24	05-22-24		1	-1	ASM4022_636_85619-LF
25195				REAR RAIL ASSY	05-24-24	05-22-24	1		0	
25196				REAR RAIL ASSY	02-14-25	02-17-25	6		6	
25197				REAR RAIL ASSY	02-17-25	02-17-25		1	5	ASM4022_636_85619-LF
25198				REAR RAIL ASSY	02-17-25	02-17-25		1	4	ASM4022_636_85619-LF
25199				REAR RAIL ASSY	02-17-25	02-17-25		1	3	ASM4022_636_85619-LF
25200				REAR RAIL ASSY	02-17-25	02-17-25		1	2	ASM4022_636_85619-LF
25201				REAR RAIL ASSY	02-17-25	02-17-25		1	1	ASM4022_636_85619-LF
25202				REAR RAIL ASSY	02-17-25	02-17-25		1	0	ASM4022_636_85619-LF
25203			**Sum**				**7**	**7**		
25204			ASM0051_240_01011-LF	FUSE BOX ASSEMBLY	02-07-25	02-10-25	5		5	
25205				FUSE BOX ASSEMBLY	02-10-25	02-10-25		5	0	ASM0051_240_00451-LF
25206			**Sum**				**5**	**5**		
25207			ASM0051_240_01301-P	GVRB WIRE SET	Past	Past	19		19	
25208			**Sum**				**19**	**0**		
25209			ASM0051_240_01431-P	NXE3350_WIRE SET TWENTY FOI	Past	Past	123		123	
25210				NXE3350_WIRE SET TWENTY FOI	05-07-24	05-07-24		6	117	ASM4022_640_82654-LF
25211				NXE3350_WIRE SET TWENTY FOI	05-08-24	05-08-24		5	112	ASM4022_640_82654-LF
25212				NXE3350_WIRE SET TWENTY FOI	05-08-24	05-08-24		5	107	ASM4022_640_82654-LF
25213				NXE3350_WIRE SET TWENTY FOI	05-08-24	05-08-24		5	102	ASM4022_640_82654-LF
25214				NXE3350_WIRE SET TWENTY FOI	05-08-24	05-08-24		5	97	ASM4022_640_82654-LF
25215				NXE3350_WIRE SET TWENTY FOI	05-14-24	05-14-24		5	92	ASM4022_640_82654-LF
25216				NXE3350_WIRE SET TWENTY FOI	05-14-24	05-14-24		4	88	ASM4022_640_82654-LF
25217				NXE3350_WIRE SET TWENTY FOI	05-14-24	05-14-24		6	82	ASM4022_640_82654-LF
25218				NXE3350_WIRE SET TWENTY FOI	05-21-24	05-21-24		1	81	ASM4022_640_82654-LF
25219				NXE3350_WIRE SET TWENTY FOI	05-22-24	05-22-24		5	76	ASM4022_640_82654-LF
25220				NXE3350_WIRE SET TWENTY FOI	05-24-24	05-24-24		5	71	ASM4022_640_82654-LF
25221				NXE3350_WIRE SET TWENTY FOI	05-29-24	05-29-24		4	67	ASM4022_640_82654-LF
25222				NXE3350_WIRE SET TWENTY FOI	06-05-24	06-05-24		4	63	ASM4022_640_82654-LF
25223				NXE3350_WIRE SET TWENTY FOI	06-05-24	06-05-24		3	60	ASM4022_640_82654-LF
25224				NXE3350_WIRE SET TWENTY FOI	06-12-24	06-12-24		5	55	ASM4022_640_82654-LF
25225				NXE3350_WIRE SET TWENTY FOI	06-14-24	06-14-24		5	50	ASM4022_640_82654-LF
25226				NXE3350_WIRE SET TWENTY FOI	06-19-24	06-19-24		5	45	ASM4022_640_82654-LF
25227				NXE3350_WIRE SET TWENTY FOI	06-26-24	06-26-24		6	39	ASM4022_640_82654-LF
25228				NXE3350_WIRE SET TWENTY FOI	06-26-24	06-26-24		5	34	ASM4022_640_82654-LF
25229				NXE3350_WIRE SET TWENTY FOI	06-28-24	07-02-24	26		60	
25230				NXE3350_WIRE SET TWENTY FOI	07-03-24	07-03-24		6	54	ASM4022_640_82654-LF
25231				NXE3350_WIRE SET TWENTY FOI	07-03-24	07-03-24		6	48	ASM4022_640_82654-LF
25232				NXE3350_WIRE SET TWENTY FOI	07-08-24	07-08-24		5	43	ASM4022_640_82654-LF

Figure 64: Screenshot of data from the Time-Phased Planning Report coming from Rapid Response

Appendix E – Put Data from PPG

A	B	C	D	E	F	H	I	J	K	L	M	N	P	Q	R	S	T	U	V
Master-order	Material	Lot	Order line	Type	Qualifica	Requested quantity	Confirmed quantity	Deviated quantity	Host's identificati	Motive	Pick-Location	Put-Location	Creation date	Quantity	Work-ord	Export Sta	Upload d	Line Info 5	Pick-Bin
ANZ001250	ASM4022_438_34242-LF	ANZ001250-2317-(	1	3	A0000N	13	13	0	80*ANZ001250*1*1	0		SH09-036-007	44931.3061	13	TR09	4	44931.31	HClass - NO	B003-1371@19
ANZ001321	ASM4022_438_34916-LF	ANZ001321-2317-(	1	3	A0000N	7	7	0	80*ANZ001321*5*1	0		SH08-003-012	44931.30738	7	TR09	4	44931.31	HClass - NO	B001-4327@5
ANZ001204	ASM4022_438_50765-LF	ANZ001204-2317-(	1	3	A0000N	4	4	0	80*ANZ001204*4*8	0		SH07-005-010	44931.30781	4	TR09	4	44931.31	HClass - NO	B002-0223@9
ANZ001166	ASM4022_438_50767-LF	ANZ001166-2317-(	1	3	A0000N	6	6	0	80*ANZ001166*1*4	0		SH08-006-018	44931.3081	6	TR09	4	44931.31	HClass - NO	B001-1109@6
ANZ001111	ASM4022_438_50769-LF	ANZ001111-2317-(	1	3	A0000N	5	5	0	80*ANZ001111*3*7	0		SH07-009-005	44931.30865	5	TR09	4	44931.31	HClass - NO	B002-2923@16
ANZ001301	ASM4022_438_32695-LF	ANZ001301-2317-(	1	3	A0000N	41	41	0	80*ANZ001301*1*1	0		SH08-007-009	44931.30904	41	TR09	4	44931.31	HClass - NO	B001-0214@7
ANZ003985	ASM4022_438_20994-LF	ANZ003985-2317-(	2	3	A0000N	20	20	0	80*ANZ003985*1*2	0		SH08-011-024	44931.30955	20	TR09	4	44931.31	HClass - NO	B006-0018@14
ANZ004452	ASM4022_478_00018-LF	ANZ004452-2317-(	1	3	A0000N	4	4	0	80*ANZ004452*2*3	0		SH01-010-02-	44931.31039	4	TR09	4	44931.31	HClass - NO	B002-2060@17
ANZ003835	ASM4022_634_22661-LF	ANZ003835-2317-(	1	3	A0000N	60	60	0	80*ANZ003835*3*1	0		SH02-042-004	44931.31081	60	TR09	4	44931.31	HClass - NO	B003-2364@21
ANZ001094	ASM4022_438_42563-LF	ANZ001094-2317-(	1	3	A0000N	20	20	0	80*ANZ001094*1*2	0		SH02-042-009	44931.3111	20	TR09	4	44931.31	HClass - NO	B003-1741@22
ANZ003985	ASM4022_438_20996-LF	ANZ003985-2317-(	1	3	A0000N	20	20	0	80*ANZ003985*1*1	0		SH01-039-01-	44931.31132	20	TR09	4	44931.31	HClass - NO	B006-0464@15
ANZ003068	ASM4022_478_00735-LF	ANZ003068-2317-(	1	3	A0000N	10	10	0	80*ANZ003068*9*9	0		SH04-005-001	44931.31252	10	TR09	4	44931.31	HClass - NO	B002-1880@8
ANZ004640	ASM4022_478_00718-LF	ANZ004640-2317-(	1	3	A0000N	2	2	0	80*ANZ004640*5*8	0		SH06-003-016	44931.31346	2	TR09	4	44931.31	HClass - NO	B001-4329@4
ANZ001254	ASM4022_438_34323-LF	ANZ001254-2317-(	1	3	A0000N	2150	2150	0	80*ANZ001254*1*2	0		SH05-010-001	44931.31417	2150	TR09	4	44931.31	HClass - NO	B002-2830@18
ANZ001337	ASM4022_438_42706-LF	ANZ001337-2317-(	1	3	A0000N	12	12	0	80*ANZ001337*6*6	0		SH06-008-001	44931.31446	12	TR09	4	44931.32	HClass - NO	B002-1229@10
ANZ004169	ASM4022_438_80530-LF	ANZ004169-2317-(	1	3	A0000N	25	25	0	80*ANZ004169*3*6	0		SH06-008-004	44931.31473	25	TR09	4	44931.32	HClass - NO	B002-2652@11
ANZ001250	ASM4022_438_34328-LF	ANZ001250-2317-(	2	3	A0000N	650	650	0	80*ANZ001250*1*2	0		SH05-030-005	44931.31541	650	TR09	4	44931.32	HClass - NO	B004-1370@23
ANZ001037	ASM4022_478_00095-LF	ANZ001037-2317-(	1	3	A0000N	9	9	0	80*ANZ001037*1*9	0		SH06-009-003	44931.3158	9	TR09	4	44931.32	HClass - NO	B002-0822@12
ANZ002824	ASM4022_478_00325-LF	ANZ002824-2317-(	1	3	A0000N	6	6	0	80*ANZ002824*3*6	0		SH06-009-007	44931.31613	6	TR09	4	44931.32	HClass - NO	B002-1525@13
ANZ001180	ASM4022_438_50728-LF	ANZ001180-2317-(	1	3	A0000N	80	80	0	80*ANZ001180*19*	0		SH06-030-005	44931.31851	248	TR09	4	44931.32	HClass - NO	B004-0894@24
ANZ002172	ASM4022_476_23141-LF	ANZ002172-2317-(	1	3	A0000N	25	25	0	80*ANZ002172*1*1	0		SH06-031-003	44931.3194	25	TR09	4	44931.32	HClass - NO	B004-0653@25
ANZ006472	ASM4022_438_25075-LF	ANZ006472-2317-(	3	3	A0000N	20	20	0	80*ANZ006472*2*4	0		SH06-039-008	44931.31972	20	TR09	4	44931.32	HClass - NO	B003-1994@20
ANZ006446	THL5962-9205803MYA-C	ANZ006446-2317-(	2	3	A0000N	162	162	0	80*ANZ006446*6*4	0		SH10-015-003	44931.3503	162	TRANUS	4	44931.35	HClass - NO	B004-0297@1
ZP0000107	AIB7001_246_06646	ZP0000107	1	3	A0000N	9	9	0	50*ZP0000107*4*0	0		SH10-009-010	44931.42653	9	TR07	4	44931.43	HClass - NO	B003-0752@4
ZP0000237	AIB7001_240_09433	ZP0000237	1	3	A0000N	24	24	0	50*ZP0000237*3*0	0		SH09-004-005	44931.42663	24	TR07	4	44931.43	HClass - NO	B002-2687@3
ZP0000265	AIB7001_246_06646	ZP0000265	1	3	A0000N	14	14	0	50*ZP0000265*3*0	0		SH10-011-003	44931.42693	14	TR07	4	44931.43	HClass - NO	B003-2051@5
MP0000081	AIB7001_240_08826	MP0000081	1	3	A0000N	40	40	0	50*MP0000081*1*C	0		SH09-027-003	44931.42728	40	TR07	4	44931.43	HClass - NO	B005-0116@2
ZP0000239	ASM4022_634_43435-LF	ZP0000239	1	3	A0000N	22	22	0	50*ZP0000239*1*0	0		SH10-020-004	44931.42796	22	TR07	4	44931.43	HClass - NO	B004-1131@11
ZP0000224	ASM4022_634_14744-LF	ZP0000224	1	3	A0000N	1	1	0	50*ZP0000224*5*0	0		SH07-014-002	44931.42912	1	TR07	4	44931.43	HClass - NO	B003-3116@8
ZP0000373	ASM4022_634_91201-LF	ZP0000373	1	3	A0000N	4	4	0	50*ZP0000373*3*0	0		SH08-022-004	44931.42942	4	TR07	4	44931.43	HClass - NO	B004-1113@12
ZP0000121	ASM4022_634_14744-LF	ZP0000121	1	3	A0000N	2	2	0	50*ZP0000121*1*0	0		SH02-016-007	44931.42971	2	TR07	4	44931.43	HClass - NO	B003-0949@10
ZP0000096	AIB7001_246_05595	ZP0000096	1	3	A0000N	32	32	0	50*ZP0000096*1*0	0		SH01-017-03-	44931.43003	32	TR07	4	44931.43	HClass - NO	B003-438@13
ZP0000450	ASM4022_682_04361-LF	ZP0000450	1	3	A0000N	1	1	0	50*ZP0000450*1*0	0		SH01-039-06-	44931.43065	1	TR07	4	44931.43	HClass - NO	B006-0458@1
ZP0000108	AIB7001_246_05547	ZP0000108	1	3	A0000N	4	4	0	50*ZP0000108*1*0	0		SH03-005-001	44931.43131	4	TR07	4	44931.43	HClass - NO	B002-2282@6
ZP0000227	ASM4022_634_83312-LF	ZP0000227	1	3	A0000N	9	9	0	50*ZP0000227*8*0	0		SH05-013-005	44931.44557	9	TR07	4	44931.45	HClass - NO	B003-0604@7
ZP0000043	ASM4022_634_14744-LF	ZP0000043	1	3	A0000N	1	1	0	50*ZP0000043*3*0	0		SH05-015-004	44931.44587	1	TR07	4	44931.45	HClass - NO	B003-2617@9
ANZ005575	ASM4022_671_89261-LF	ANZ005575-2319-(	2	3	A0000N	11	11	0	80*ANZ005575*13*	0		SH10-016-002	44932.3962	30	TRSPOED	4	44932.4	HClass - NO	B004-0508@1
ANZ004834	ASM4022_640_90481-LF	ANZ004834-2321-(	3	3	A0000N	20	20	0	80*ANZ004834*5*6	0		SH09-020-005	44932.40154	20	TRSPOED	4	44932.4	HClass - NO	B004-1509@3
ANP000453	ASM4022_478_00806-LF	ANP000453-2321-(	1	3	A0000N	4	4	0	80*ANP000453*4*2	0		SH10-010-009	44932.40178	4	TRSPOED	4	44932.4	HClass - NO	B003-2233@4
ANZ004584	ASM4022_640_98631-LF	ANZ004584-2321-(	3	3	A0000N	193	193	0	80*ANZ004584*3*4	0		SH10-017-001	44932.40269	193	TRSPOED	4	44932.4	HClass - NO	B004-1554@2

Figure 65: Screenshot of a list with Put Data coming from PPG

Appendix F – Impression of Box Capacity Usage

1	Material	Location	Bin	Put date	Current quantity	Max Quantity	Box Capacity Usage		Average Box Capacity Usage
2	ASM4022_711_21431-LF	SH07-030-004-01	B003	2024-04-08	8	10	80.0%		64.8%
3	ASM4022_685_71132-LF	SH04-021-002-01	B003	2024-04-23	15	15	100.0%		
4	ASM4022_476_23022-LF	SH05-040-010-02	B003	2024-02-27	8	10	80.0%		
5	ASM4022_476_01401-LF	SH08-014-010-01	B003	2024-03-22	6	13	46.2%		
6	ASM4022_660_28752-LF	SH01-009-08-01	B002	2023-03-31	1	2	50.0%		
7	ASM4022_698_29811-LF	SH04-007-003-01	B002	2024-04-10	8	8	100.0%		
8	ASM4022_472_65202-LF	SH02-004-010-03	B002	2023-06-22	10	42	23.8%		
9	ASM4022_438_41288-LF	SH07-011-003-03	B001	2023-03-31	1	10	10.0%		
10	ASM4022_438_40734-LF	SH06-005-007-03	B002	2024-02-15	11	20	55.0%		
11	ASM4022_438_33382-LF	SH08-007-009-02	B001	2023-03-31	53	72	73.6%		
12	ASM4022_478_00740-LF	SH07-002-003-01	B001	2024-04-15	4	10	40.0%		
13	ASM4022_693_97832-LF	SH09-030-034-01	B006	2024-04-03	4	10	40.0%		
14	ASM4022_439_91043-LF	SH05-003-012-03	B001	2023-03-31	94	200	47.0%		
15	ASM4022_438_50506-LF	SH05-017-004-02	B003	2023-11-09	3	37	8.1%		
16	ASM4022_685_40962-LF	SH09-032-002-02	B004	2024-01-17	13	23	56.5%		
17	ASM4022_478_00553-LF	SH10-009-007-02	B003	2024-03-20	14	19	73.7%		

Figure 66: Impression of the Box Capacity Usage Calculation

Appendix G – Proposed Reallocations by the Box Division Model

SKU	Current quantity	Location	Storage unit	Carrier	Bin	PreferredBox	RequiredBoxes
ASM4022_478_00622-LF	2	SH08-031-003-01	SH08	31	B004	B001	2
ASM4022_671_13681-LF	9	SH09-019-005-01	SH09	19	B004	B006	2
ASM4022_668_83022-LF	12	SH05-031-002-02	SH05	31	B004	B006	3
ASM4022_668_83022-LF	16	SH02-020-004-01	SH02	20	B004	B006	4
ASM4022_478_00101-LF	139	SH04-024-004-01	SH04	24	B004	B002	2
ASM4022_478_01265-LF	340	SH09-022-005-01	SH09	22	B004	B002	2
ASM4022_478_01265-LF	270	SH07-022-004-02	SH07	22	B004	B002	2
ASM4022_668_82922-LF	8	SH09-025-003-01	SH09	25	B004	B002	2
ASM4022_640_91023-LF	2	SH05-031-002-01	SH05	31	B004	B002	2
ASM4022_668_95531-LF	20	SH09-020-001-02	SH09	20	B004	B002	2
ASM4022_646_79621-LF	13	SH04-022-005-02	SH04	22	B004	B002	2
ASM4022_677_24701-P	9	SH01-029-05-01	SH01	29	B004	B002	2
ASM4022_668_34262-LF	20	SH09-033-005-01	SH09	33	B004	B002	4
ASM4022_478_00490-LF	120	SH10-025-003-02	SH10	25	B004	B002	4
ASM4022_438_34090-LF	135	SH06-024-002-02	SH06	24	B004	B002	4
ASM4022_668_34262-LF	20	SH09-020-003-01	SH09	20	B004	B002	4
ASM4022_711_86641-LF	3	SH09-016-001-01	SH09	16	B004	B003	2
ASM4022_634_82011-LF	16	SH07-028-002-02	SH07	28	B004	B003	2
ASM4022_685_87024-LF	30	SH09-023-005-01	SH09	23	B004	B003	2

Figure 67: Overview of proposed OTM reallocations by the box division model

Appendix H - Pseudocode of the First Step in the Reallocation Model

```
# Define the shuttles and pickers
Shuttles = ["SH01", "SH02", "SH03", "SH04", "SH05", "SH06", "SH07", "SH08", "SH09", "SH10"]
Pickers = ["Picker1", "Picker2", "Picker3", "Picker4", "Picker5"]

# List of reallocations that must be extracted per shuttle
Reallocations = {
    "SH01": [], "SH02": [], "SH03": [], "SH04": [],
    "SH05": [], "SH06": [], "SH07": [], "SH08": [],
    "SH09": [], "SH10": []
}

# Initialize picker reallocation lists
PickerList = {
    "Picker1": [], "Picker2": [], "Picker3": [],
    "Picker4": [], "Picker5": []
}

# Overview of dedicated shuttles per picker
DedicatedShuttles = {
    "Picker1": ["SH01", "SH02"], "Picker2": ["SH03", "SH04"],
    "Picker3": ["SH05", "SH06"], "Picker4": ["SH07", "SH08"],
    "Picker5": ["SH09", "SH10"]
}

# Time Parameters
PurgeTime = 10  # seconds
TempStoreTime = 30  # seconds
CarrierTravelTime = 25  # seconds

# Function to calculate picking time for two shuttles
def CalculatePickingTimeTwoShuttles():
    return PurgeTime + TempStoreTime

# Function to calculate picking time for one shuttle
def CalculatePickingTimeOneShuttle():
    return PurgeTime + TempStoreTime + CarrierTravelTime

# Saves total picking time per order picker
TotalPickingTimes = {}

for Picker in Pickers:
    # Determine dedicated shuttles
    StartingShuttle = DedicatedShuttles[Picker][0]
    SecondShuttle = DedicatedShuttles[Picker][1]

    # Get reallocations from both shuttles
    Shuttle1Reallocations = Reallocations[StartingShuttle]
    Shuttle2Reallocations = Reallocations[SecondShuttle]

    # Variable to track total picking time
    TotalPickingTime = 0

    # Initially add one task from each shuttle to fill the picker list with two tasks
    if Shuttle1Reallocations:
        PickerList[Picker].append(Shuttle1Reallocations.pop(0))
    if Shuttle2Reallocations:
        PickerList[Picker].append(Shuttle2Reallocations.pop(0))

    # Calculate time for extracting reallocation while using both shuttles
    while Shuttle1Reallocations and Shuttle2Reallocations:
        PickerList[Picker].append(Shuttle1Reallocations.pop(0))
        PickerList[Picker].append(Shuttle2Reallocations.pop(0))
        TotalPickingTime += CalculatePickingTimeTwoShuttles()  # Adds time

    # Extract reallocations from the remaining shuttle and calculate time of completion
    if Shuttle1Reallocations:  # If Shuttle1 has remaining tasks
        while Shuttle1Reallocations:
            PickerList[Picker].append(Shuttle1Reallocations.pop(0))
            TotalPickingTime += CalculatePickingTimeOneShuttle()  # Adds time
    elif Shuttle2Reallocations:  # If Shuttle2 has remaining tasks
        while Shuttle2Reallocations:
            PickerList[Picker].append(Shuttle2Reallocations.pop(0))
            TotalPickingTime += CalculatePickingTimeOneShuttle()  # Adds time

    # Store the total picking time for the picker
    TotalPickingTimes[Picker] = TotalPickingTime

# Print reallocation overview and total picking times
print("Reallocation List:", PickerList)
print("Total Picking Times (sec):", TotalPickingTimes)
```

Appendix I - Pseudocode for the Second Step in the Reallocation Model

```
# Define the shuttles and pickers
Shuttles = ["SH01", "SH02", "SH03", "SH04", "SH05", "SH06", "SH07", "SH08", "SH09", "SH10"]
Pickers = ["Picker1", "Picker2", "Picker3", "Picker4", "Picker5"]

# List of proposed carrier type changes for each shuttle (carrier number, current box type, occupied boxes,
new carrier type)
CarrierChanges = {
    "SH01": [("Carrier1", "B004", 5, "B003")],
    "SH02": [],
    "SH03": [],
    "SH04": [],
    "SH05": [],
    "SH06": [],
    "SH07": [],
    "SH08": [],
    "SH09": [],
    "SH10": []
}

# Carrier capacity (boxes available per carrier type)
CarrierCapacity = {"B001": 80, "B002": 40, "B003": 20, "B004": 10, "B005": 10, "B006": 22}

# Initialize picker job lists for two dedicated shuttles
PickerJobs = {picker: {"FirstShuttle": [], "SecondShuttle": []} for picker in Pickers}

# Overview of dedicated shuttles per picker
DedicatedShuttles = {
    "Picker1": ["SH01", "SH02"],
    "Picker2": ["SH03", "SH04"],
    "Picker3": ["SH05", "SH06"],
    "Picker4": ["SH07", "SH08"],
    "Picker5": ["SH09", "SH10"]
}

# Define time parameters time (in seconds)
PurgeTime = 10
TempStoreTime = 35
ClearCarrierTime = 10
BoxesToRemoveTime = 10
EmptyBinRemovalTime = 10
FillCarrierTime = 10
EmptyBinStoreTime = 10
CarrierTravelTimeDefault = 25

# Function to calculate the time for carrier type changes with a conditional check for carrier travel time
def CalculateCarrierChangeTime(OccupiedBoxes, BoxesToRemove, BoxesToAdd, BothShuttlesAvailable):
    # Only add CarrierTravelTime if jobs are only available in one shuttle
    CarrierTravelTime = CarrierTravelTimeDefault if not BothShuttlesAvailable else 0

    RemovalTime = OccupiedBoxes * (PurgeTime + TempStoreTime) + ClearCarrierTime + BoxesToRemove *
EmptyBinRemovalTime

    AdditionTime = FillCarrierTime + BoxesToAdd * EmptyBinStoreTime

    return CarrierTravelTime + RemovalTime + AdditionTime

# Function to assign carrier changes to pickers and calculate the total time
def ProcessCarrierChanges():
    TotalTimePerPicker = {picker: 0 for picker in Pickers}

    for Picker in Pickers:
        # Get the two dedicated shuttles for the picker
        FirstShuttle, SecondShuttle = DedicatedShuttles[Picker]

        # Split the jobs into two lists for the two shuttles
        FirstShuttleJobs = CarrierChanges.get(FirstShuttle, [])
        SecondShuttleJobs = CarrierChanges.get(SecondShuttle, [])

        # Assign one job from each shuttle to the picker initially
        if FirstShuttleJobs:
            PickerJobs[Picker]["FirstShuttle"].append(FirstShuttleJobs.pop(0))
        if SecondShuttleJobs:
            PickerJobs[Picker]["SecondShuttle"].append(SecondShuttleJobs.pop(0))

        # Alternate between the two shuttles until one shuttle is empty
        while FirstShuttleJobs or SecondShuttleJobs:
            BothShuttlesAvailable = bool(FirstShuttleJobs and SecondShuttleJobs)  # Check if both shuttles
have jobs

            # Extract a task from the first shuttle if available
            if FirstShuttleJobs:
                PickerJobs[Picker]["FirstShuttle"].append(FirstShuttleJobs.pop(0))
```

```
                CurrentJob = PickerJobs[Picker]["FirstShuttle"][-1]  # Last job of the list
                CurrentCarrierType = CurrentJob[1]
                NewCarrierType = CurrentJob[3]
                OccupiedBoxes = CurrentJob[2]

                # Calculate boxes to add
                BoxesToAdd = CarrierCapacity[NewCarrierType]
                BoxesToRemove = OccupiedBoxes

                # Calculate time for the task and add to the total time
                TimeForChange = CalculateCarrierChangeTime(OccupiedBoxes, BoxesToRemove, BoxesToAdd,
BothShuttlesAvailable)
                TotalTimePerPicker[Picker] += TimeForChange

            # Extract a task from the second shuttle if available
            if SecondShuttleJobs:
                PickerJobs[Picker]["SecondShuttle"].append(SecondShuttleJobs.pop(0))
                CurrentJob = PickerJobs[Picker]["SecondShuttle"][-1]
                CurrentCarrierType = CurrentJob[1]
                NewCarrierType = CurrentJob[3]
                OccupiedBoxes = CurrentJob[2]

                # Calculate boxes to add
                BoxesToAdd = CarrierCapacity[NewCarrierType]
                BoxesToRemove = OccupiedBoxes

                # Calculate time for the task and add to the total time
                TimeForChange = CalculateCarrierChangeTime(OccupiedBoxes, BoxesToRemove, BoxesToAdd,
BothShuttlesAvailable)
                TotalTimePerPicker[Picker] += TimeForChange

    return TotalTimePerPicker

# Run the carrier change process and display results
ResultTimes = ProcessCarrierChanges()
print("Total time per picker:", ResultTimes)
print("Schedule for Pickers",PickerJobs)
```

Appendix J - Pseudocode for the Third Step in the Reallocation Model

```
# Variables
CostRatePerMinute = 1  # Cost rate per minute
Iterations = 1000  # Number of iterations
PurgeTime = 10  # Time for purge operation in OTM reallocation
PrintTime = 30  # Time for printing labels in OTM reallocation
Shuttles = ["SH01", "SH02", "SH03", "SH04", "SH05", "SH06", "SH07", "SH08", "SH09", "SH10"]
ShuttleTravelTimes[Shuttle1][Shuttle2] # Matrix with travel times between all snuttles

# Carrier capacity (number of boxes available for storage) for each shuttle and each box type
CarrierCapacity = {
    "SH01": {"B001": 80, "B002": 40, "B003": 20, "B004": 10, "B005": 2,"B006": 5},  # Example for SH01
    "SH02": {"B001": 60, "B002": 60, "B003": 40, "B004": 10, "B005": 6,"B006": 2},  # Example for SH02
    "SH03": {}, "SH04": {}, "SH05": {}, "SH06": {}, "SH07": {}, "SH08": {}, "SH09": {}, "SH10": {}
    # SH03 to SH10 also have capacities simlar to SH01 and SH02 but are left empty for brevity
}

# Job lists for Extracted OTM and OTM Reallocations
# Job info contains (Carrier, CurrentBoxType, RecommendedBoxType, RequiredBoxes)
ExtractedOTMJobs = {
    "SH01": [("Carrier1", "B004", "B002", 2)],
    "SH02": [], "SH03": [], "SH04": [], "SH05": [], "SH06": [], "SH07": [], "SH08": [], "SH09": [], "SH10":
[]
}

OTMJobs = {
    "SH01": [("Carrier2", "B002", "B001", 3)],
    "SH02": [], "SH03": [], "SH04": [], "SH05": [], "SH06": [], "SH07": [], "SH08": [], "SH09": [], "SH10":
[]
}

Pickers = ["Picker1", "Picker2", "Picker3", "Picker4", "Picker5"]
JobList = {Picker: [] for Picker in Pickers}

# Initialize total time counters for Extracted OTM and OTM
TotalTimeExtractedOTM = 0
TotalTimeOTM = 0

# Initialize completion time tracking for each picker
PickerCompletionTimes = {Picker: 0 for Picker in Pickers}

# Function to get the picker with the least current completion time
def GetPickerWithLeastCompletionTime():
    return min(PickerCompletionTimes, key=PickerCompletionTimes.get)

# Function to calculate completion time for 2 boxes
def OTMCompletionTimeForTwoBoxes(CurrentShuttle, PickLocation, PutLocation):
    StorePrep = 20  # As StorePrep = 10 * RequiredBoxes (2 boxes)
    PickTravelTime = ShuttleTravelTimes[CurrentShuttle][PickLocation]
    PutTravelTime = ShuttleTravelTimes[PickLocation][PutLocation]
    BoxPickupTime = random.randint(10, 20)
    CarrierTravelTime = 25
    PutTime = random.randint(10, 20)

    RemovalTime = PickTravelTime + BoxPickupTime
    AdditionTime = PutTravelTime + CarrierTravelTime + PutTime
    return StorePrep + RemovalTime + AdditionTime

# Function to calculate completion time for more than 2 boxes
def OTMCompletionTimeForMoreThanTwoBoxes(CurrentShuttle, PickLocation, PutLocation, RequiredBoxes):
    StorePrep = 10 * RequiredBoxes
    PickTravelTime = ShuttleTravelTimes[CurrentShuttle][PickLocation]
    PutTravelTime = ShuttleTravelTimes[PickLocation][PutLocation]
    CarrierTravelTime = 25
    PutTime = random.randint(5 * RequiredBoxes, 10 * RequiredBoxes)

    RemovalTime = PickTravelTime + random.randint(10, 20)  # TravelTime + BoxPickupTime
    AdditionTime = PutTravelTime + CarrierTravelTime + PutTime
    return PrintTime + StorePrep + RemovalTime + AdditionTime

# Function to simulate OTM completion time
def SimulateOTMCompletionTime(CurrentShuttle, PickLocation, PutLocation, RequiredBoxes):
    if RequiredBoxes == 2:
        return OTMCompletionTimeForTwoBoxes(CurrentShuttle, PickLocation, PutLocation)
    else:
        return OTMCompletionTimeForMoreThanTwoBoxes(CurrentShuttle, PickLocation, PutLocation,
RequiredBoxes)

# Function to assign a random put location on a different shuttle
def GetRandomPutLocation(CurrentShuttle, NewBoxType, RequiredBoxes):
    AvailableShuttles = [Shuttle for Shuttle in Shuttles if Shuttle != CurrentShuttle]

    # Loop until a shuttle with enough space is found
    while AvailableShuttles:
        PutLocation = random.choice(AvailableShuttles)
```

```
        # Check if the selected shuttle has enough capacity for the required box type
        if CarrierCapacity[PutLocation].get(NewBoxType, 0) >= RequiredBoxes:
            return PutLocation
        else:
            # Remove the current shuttle from the available list and try again
            AvailableShuttles.remove(PutLocation)


# Simulate Extracted OTM Jobs
for Iteration in range(Iterations):
    for Shuttle, JobListForShuttle in ExtractedOTMJobs.items():
        for Job in JobListForShuttle:
            Carrier, BoxType, NewBoxType, RequiredBoxes = Job
            PickLocation = Shuttle  # Assuming picking happens at the current shuttle
            PutLocation = GetRandomPutLocation(Shuttle, NewBoxType, RequiredBoxes)
            if PutLocation:
                CompletionTime = SimulateOTMCompletionTime(Shuttle, PickLocation, PutLocation,
RequiredBoxes)

                # Assign the job to the picker with the least current completion time
                Picker = GetPickerWithLeastCompletionTime()

                # Update the CarrierCapacity for the new box type
                CarrierCapacity[PutLocation][NewBoxType] -= RequiredBoxes

                JobList[Picker].append((Carrier, BoxType, NewBoxType, CompletionTime, PutLocation))
                TotalTimeExtractedOTM += CompletionTime

                # Update the picker's completion time
                PickerCompletionTimes[Picker] += CompletionTime

# Simulate OTM Jobs
for Iteration in range(Iterations):
    for Shuttle, JobListForShuttle in OTMJobs.items():
        for Job in JobListForShuttle:
            Carrier, BoxType, NewBoxType, RequiredBoxes = Job
            PickLocation = Shuttle
            PutLocation = GetRandomPutLocation(Shuttle, NewBoxType, RequiredBoxes)
            if PutLocation:
                CompletionTime = SimulateOTMCompletionTime(Shuttle, PickLocation, PutLocation,
RequiredBoxes) + PurgeTime

                # Assign the job to the picker with the least current completion time
                Picker = GetPickerWithLeastCompletionTime()

                # Update the CarrierCapacity for the new box type
                CarrierCapacity[PutLocation][NewBoxType] -= RequiredBoxes

                JobList[Picker].append((Carrier, BoxType, NewBoxType, CompletionTime, PutLocation))
                TotalTimeOTM += CompletionTime

                # Update the picker's completion time
                PickerCompletionTimes[Picker] += CompletionTime

# Average time for each category
AvgTimeExtractedOTM = TotalTimeExtractedOTM / Iterations
AvgTimeOTM = TotalTimeOTM / Iterations

# Total cost calculations
TotalCostExtractedOTM = (AvgTimeExtractedOTM / 60) * CostRatePerMinute
TotalCostOTM = (AvgTimeOTM / 60) * CostRatePerMinute
TotalCost = TotalCostExtractedOTM + TotalCostOTM

# Output
print("Total cost for Extracted OTM reallocations:", TotalCostExtractedOTM, "euros")
print("Total cost for OTM reallocations:", TotalCostOTM, "euros")
print("Overall total cost:", TotalCost, "euros")
```

Appendix K - Pseudocode for the OTO Nearest Neighbour Heuristic

```
# Initialize the completion times for each order picker to 0
for Picker in OrderPickerTimes:
    OrderPickerTimes[Picker] = 0

# Initialize the availability time for each shuttle to 0
for Shuttle in ShuttleAvailability:
    ShuttleAvailability[Shuttle] = 0

# Initialize the schedule and current location for each order picker
OrderPickers = ['Picker1', 'Picker2', 'Picker3', 'Picker4', 'Picker5']
InitialLocations = ['SH01', 'SH03', 'SH05', 'SH07', 'SH09']

for i in range(len(OrderPickers)):
    Picker = OrderPickers[i]
    OrderPickersSchedule[Picker] = []                   # Each picker has an empty schedule to start with
    OrderPickerCurrentLocation[Picker] = InitialLocations[i]  # Set initial location

# Function to find the picker with the minimum completion time
def FindUnoccupiedPicker(OrderPickerTimes):
    return min(OrderPickerTimes, key=OrderPickerTimes.get)  # Returns the picker with the minimum
completion time

# Start iterating over each OTO reallocation job
for Job in OTOReallocations:
    RecommendedBoxType = Job['Bin']        # Recommended new box type
    JobShuttle = Job['Storage unit']       # Current location of the original box
    RequiredBoxes = 1                      # Always 1 box required

    # Select the picker with the minimum completion time
    FreePicker = FindUnoccupiedPicker(OrderPickerTimes)
    CurrentShuttle = OrderPickerCurrentLocation[FreePicker]

    # Find the nearest shuttle with space for 1 box of the specified type
    PickShuttle = JobShuttle
    PutShuttle = FindNearestShuttleWithSpace(RequiredBoxes, RecommendedBoxType, ExcludeShuttle=PickShuttle)

    # Calculate pick and put times
    PickTime = CalculatePickTime(Job, FreePicker, CurrentShuttle, PickShuttle)
    PutTime = CalculatePutTime(Job, FreePicker, CurrentShuttle, PutShuttle)

    # Determine start and end times, ensuring no overlap with other tasks or shuttle usage
    StartTime = max(OrderPickerTimes[FreePicker], ShuttleAvailability[PickShuttle],
ShuttleAvailability[PutShuttle])
    EndTime = StartTime + PickTime + PutTime

    # Update shuttle availability for the pick and put locations, releasing after each operation
    ShuttleAvailability[PickShuttle] = StartTime + PickTime     # Free PickShuttle after pick operation
    ShuttleAvailability[PutShuttle] = EndTime                   # Free PutShuttle after put operation

    # Assign the job to the picker and update their schedule
    OrderPickersSchedule[FreePicker].append((Job, StartTime, StartTime + PickTime, PickShuttle, 'pick'))
    OrderPickersSchedule[FreePicker].append((Job, StartTime + PickTime, EndTime, PutShuttle, 'put'))
    OrderPickerTimes[FreePicker] = EndTime
    OrderPickerCurrentLocation[FreePicker] = PutShuttle  # Update picker's location after job
```

Appendix L - Pseudocode for the OTO Farthest Neighbour Heuristic

```
# Initialize the completion times for each order picker to 0
for Picker in OrderPickerTimes:
    OrderPickerTimes[Picker] = 0

# Initialize the availability time for each shuttle to 0
for Shuttle in ShuttleAvailability:
    ShuttleAvailability[Shuttle] = 0

# Initialize the schedule and current location for each order picker
OrderPickers = ['Picker1', 'Picker2', 'Picker3', 'Picker4', 'Picker5']
InitialLocations = ['SH01', 'SH03', 'SH05', 'SH07', 'SH09']

for i in range(len(OrderPickers)):
    Picker = OrderPickers[i]
    OrderPickersSchedule[Picker] = []                    # Each picker has an empty schedule to start with
    OrderPickerCurrentLocation[Picker] = InitialLocations[i]  # Set initial location

# Function to find the picker with the minimum completion time
def FindUnoccupiedPicker(OrderPickerTimes):
    return min(OrderPickerTimes, key=OrderPickerTimes.get)  # Returns the picker with the minimum
completion time

# Start iterating over each OTO reallocation job
for Job in OTOReallocations:
    RecommendedBoxType = Job['Bin']         # Recommended new box type
    JobShuttle = Job['Storage unit']        # Current location of the original box
    RequiredBoxes = 1                       # Always 1 box required

    # Select the picker with the minimum completion time
    FreePicker = FindUnoccupiedPicker(OrderPickerTimes)
    CurrentShuttle = OrderPickerCurrentLocation[FreePicker]

    # Find the farthest shuttle with space for 1 box of the specified type
    PickShuttle = JobShuttle
    PutShuttle = FindFarthestShuttleWithSpace(RequiredBoxes, RecommendedBoxType,
ExcludeShuttle=PickShuttle)

    # Calculate pick and put times
    PickTime = CalculatePickTime(Job, FreePicker, CurrentShuttle, PickShuttle)
    PutTime = CalculatePutTime(Job, FreePicker, CurrentShuttle, PutShuttle)

    # Determine start and end times, ensuring no overlap with other tasks or shuttle usage
    StartTime = max(OrderPickerTimes[FreePicker], ShuttleAvailability[PickShuttle],
ShuttleAvailability[PutShuttle])
    EndTime = StartTime + PickTime + PutTime

    # Update shuttle availability for the pick and put locations, releasing after each operation
    ShuttleAvailability[PickShuttle] = StartTime + PickTime     # Free PickShuttle after pick operation
    ShuttleAvailability[PutShuttle] = EndTime                   # Free PutShuttle after put operation

    # Assign the job to the picker and update their schedule
    OrderPickersSchedule[FreePicker].append((Job, StartTime, StartTime + PickTime, PickShuttle, 'pick'))
    OrderPickersSchedule[FreePicker].append((Job, StartTime + PickTime, EndTime, PutShuttle, 'put'))
    OrderPickerTimes[FreePicker] = EndTime
    OrderPickerCurrentLocation[FreePicker] = PutShuttle  # Update picker's location after job
```

Appendix M - Pseudocode for the OTO Simulated Annealing Heuristic

```
# Initialize variables
CurrentSchedule = CopyInitialSchedule(InitialOrderPickerSchedule) # Schedule from constructive heuristics
CurrentTime = CalculateTotalTime(CurrentSchedule)                  # Completion time of the current schedule
CurrentTemp = InitialTemp
BestSchedule = CurrentSchedule
BestTime = CurrentTime

# Define the move operator to generate a neighbor
def MoveOperator(CurrentSchedule):
    # Perform a move, such that reallocations are adjusted in the schedule
    NewSchedule = ApplyMove(CurrentSchedule)
    return NewSchedule

# Perform the simulated annealing process
while CurrentTemp > FinalTemp:
    for Iteration in range(Iterations):
        # Use the move operator to generate a neighboring solution
        NeighborSchedule = MoveOperator(CurrentSchedule)
        NeighborTime = CalculateTotalTime(NeighborSchedule)  # Calculate total time for the new schedule

        # Accept the new schedule if it has a lower time or probabilistically if it's worse
        if NeighborTime < CurrentTime or math.exp((CurrentTime - NeighborTime) / CurrentTemp) >
random.random():
            CurrentSchedule = NeighborSchedule
            CurrentTime = NeighborTime

            # Update the best found schedule if the neighbor is better
            if NeighborTime < BestTime:
                BestSchedule = NeighborSchedule
                BestTime = NeighborTime

    # Reduce the current temperature
    CurrentTemp *= Alpha

# Return the best found schedule and time
return BestSchedule, BestTime
```