Developing a supplier timeslot indication model for an e-fulfillment center balancing incoming workload

T.H.Kerkhof

August 2023

Educational Institute
University of Twente
Faculty of Behavioural, Management and Social Sciences
Information Systems
Drienerlolaan 5
7522 NB Enschede
053 489 9111

Educational Program
Industrial Engineering and Management
Production and Logistics Management

Supervisors University of Twente
Dr. A. Trivella
Assistant Professor of Operations Research
Dep. Industrial Engineering and Business Information Systems
Faculty of Behavioural Management and Social Sciences

Dr. ir. E.A. Lalla-Ruiz
Assistant Professor in Logistics and Operations Research
Dep. Industrial Engineering and Business Information Systems
Faculty of Behavioural Management and Social Sciences

Supervisor Company X

Preface

This document presents the outcome of my graduation period during which I had the opportunity to conduct research within an online retailer, namely Company X. Over the past few months, I have been able to apply the knowledge acquired during my master's degree in Industrial Engineering and Management in a practical setting.

First, I would like to express my gratitude to the company team for welcoming me and making me feel at home within the company. They have ensured that I had a pleasant experience during this period, and their support has been of great value, particularly during challenging times.

I would also like to extend special thanks to my graduation supervisor at Company X. He facilitated introductions with key individuals within the company, provided guidance, and motivated me until the completion of my graduation research.

Additionally, I would like to acknowledge my supervisors Alessio Trivella and Eduardo Lalla from the University of Twente. I greatly appreciate our collaboration and the substantial amount of time they dedicated to assisting me. Their guidance and feedback significantly enhanced the quality of my thesis.

Enjoy reading my master's thesis.

Tim Kerkhof

Management summary

Problem context

This research is conducted at a company that wants to remain anonymous, it is referred to as Company X. Company X is one of the largest online retailers in the Netherlands and Belgium and daily they place orders for new products which are sent to their warehouses by trucks. The current arrival process of trucks delivering items to Company X's biggest warehouse, referred to as Warehouse 1, leads to significant arrival peaks in the morning compared to the rest of the day. This results in an uneven distribution of items flowing into the Warehouse 1.

After the pallets are unloaded from the trucks, they will be moved to the receiving phase where the items will be taken off the pallets. All the pallets from one shipment will be moved together towards the receiving phase and because there are significantly more shipments during the morning hours, this will cause the receiving workstations to not be able to process all of these pallets. Therefore, pallets filled with items will be placed in a waiting area between the unloading - and receiving phase, which is called the Work-In-Progress (*WIP*) area. Nowadays, the average number of items waiting in the WIP area has become so large, that it negatively affects the dock-to-stock time. This is defined as the time it takes from unloading an item until it is putaway in the stock area of the warehouse. It is measured as a metric called On-Time Performance (*OTP*) and this metric keeps track of the dock-to-stock time of an item and if it is below 72 hours. Unfortunately, because of the number of items in the WIP area, items have to wait for too long and not all items can be stocked within the 72-hour timeframe. Another performance indicator is the productivity of the unloading phase, which is negatively affected by the decrease in incoming pallets in the afternoon because operators become idle. Based on these observations, Company X wants to have more control over the incoming trucks carrying pallets, and the following research question is formulated:

"*How can long-term inbound shipment planning be used to reduce the amount of daily WIP at the inbound area of the Company X's Warehouse 1?*"

Method

Our truck scheduling problem is described in the literature as an identical parallel machine scheduling problem, where incoming trucks need to be scheduled on a number of parallel docks which all have the same processing time, depending on the number of pallets the truck is carrying. Due to the large number of suppliers for Company X, we have decided to allocate fixed long-term timeslots to a number of suppliers. With the use of our literature research, we were able to design our scheduling problem as a mixed-integer linear program. Cross-docking operations research has us decided to develop a Simulated Annealing algorithm to solve our problem because it is too large to solve to optimality. This algorithm uses historical shipment data and is able to measure the impact of allocating fixed delivery timeslots to a number of suppliers.

Experimental results

As an input, we gave the Simulated Annealing algorithm a minimum of 20 selected companies, which together count for 48.3% of the total items delivered over the 168 days of historical shipment data. In addition, we provided the option to include a maximum of 10 additional companies to the solution. This resulted in a final selection of 27 companies and a total number of items in the WIP area of 6,156,372, which is a decrease of 19.77% compared to the current situation of 7,673,278 items. Secondly, we conducted more experiments with different numbers of companies to be added to the solution and given a fixed timeslot, and the best result was found with 29 companies and a total number of 6,075,968 items (-20.82% compared to the current situation).

Conclusions and recommendations

The Simulated Annealing algorithm we created is able to solve the mathematical model that describes the identical parallel machine scheduling problem of Company X. It demonstrates how long-term inbound shipment planning, in the form of fixed timeslot agreements with suppliers, is able to reduce the number of items in the WIP area of the inbound phase of the Company X's Warehouse 1.

To further improve the existing inbound truck scheduling process, we recommend conducting additional research into the exact processing times of the inbound procedures. This will enable a more realistic estimation of which company should get which timeslot. We also recommend taking into account multi-site deliveries, because Warehouse 1 is located in the same building as the second warehouse called Warehouse 2. Therefore, the timeslot allocation of Warehouse 1 should be coordinated with Warehouse 2 to ensure that a company is not scheduled in the morning and in the afternoon on the same day for both warehouses.

Contents

List of Figures

List of Tables

Glossary of Terms and Abbreviations

HDs	Handling Devices.
GLN	Global Location Number.
KPIs	Key Performance Indicators.
MILP	Mixed Integer Linear Programming.
OTP	On-Time Performance.
SaaS	Software as a Service.
SPT	Shortest Processing Time.
VRSP	Vehicle Routing and Scheduling Problem.
WIP	Work In Progress.
WMS	Warehouse Management System.

Chapter 1

Introduction

This first chapter serves as an introduction to the research problem and provides the foundation for the remainder of this study. Section 1.1 provides a description of the company Company X, while section 1.2 explains the logistical process within and around their warehouse. Subsequently, section 1.3 elaborates on the research problem through a problem cluster analysis. Finally, section 1.4 is about the scope of the research, and section 1.5 formulates the research questions and research design.

1.1 Company description

Company X is an online retailer that sells a wide variety of online products. In recent years, Company X has not only focused on purchasing and selling items online but also on developing its functioning as an online platform. This allows third parties to use Company X's services, such as stocking items in their warehouse and shipping them to the customers. Currently, there are 4 propositions that cause item shipments:

1. Retailer 1: the first product flow is the items that are bought by Company X themselves from different suppliers and the internal department responsible for these purchases is called 'Retailer 1'.
2. Logistics via Company X: these are all the items received from the companies and entrepreneurs, also called partners, who use the service of Company X to stock their items and send them to the customers once they are ordered online via Company X's website.
3. Shipment via Company X: partners who make use of this service stock and pack their items themselves, but use Company X's website as an online shop

window. Once an order has been made, Company X takes responsibility for the shipment to the customer.

4. Plaza partners: these partners only use the website to sell their products. The process of stocking, packing, and shipping is done by themselves.

Last year Company X counted around 13 million active customers in the Netherlands and Belgium with an assortment of around 47 million articles. The online platform of Company X has already facilitated more than 50.000 partners with selling their products.

To stock all these articles and handle returns, Company X uses 6 different warehouses:

1. Warehouse 1: This in Waalwijk located warehouse started operating in 2017 and got expanded under the same roof in May 2022 with another fulfillment center. The total floor area of the building is around 100000 square meters and here the focus lies on small to medium-sized items. From an outbound perspective, both halves are seen as one merged warehouse. But from an inbound perspective, they are 2 separate warehouses with their own Global Location Number (*GLN*).
2. Warehouse 2: this facility stores medium to large-sized items.
3. Warehouse 3: this is a specialized facility based in Nieuwegein where large and extra-large items are stocked like washing machines and refrigerators.
4. Warehouse 4: from this warehouse in Culemborg most of the books are shipped. Unlike the rest of the warehouses, other companies also make use of this facility.
5. Warehouse 5: in 2021 Company X opened its own return center for customer returns of books, small -, medium - and large items.
6. Warehouse 6: this is the most recent warehouse which is optimized for mono orders of just one item and it can be used during the peak weeks in November and December.

1.2 Logistical process

Pallets and parcels are transported to the destined warehouse with the use of trucks. Once Company X places an order with the supplier or the partner decides to send items to the warehouse, the subsequent step involves organizing transportation. The supplier or the partner themselves undertake this responsibility, which falls outside the scope of Company X's daily operations.

An exception to this is Company X's First Mile service, which makes it possible for parcels to be collected from partners. Multiple shipments from the same region are then consolidated and transported to the warehouse. Around a third of the parcels from partners are being transported with the use of this service, the rest is sent by the partners themselves.

Upon the arrival of a truck carrying goods for Company X at the warehouse, it has to wait in front of a gate, communicate with the doorman regarding the nature of the shipment, and substantiate this with the right paperwork. If everything is approved, the driver is directed to a designated dock area, where the inbound process starts. This process consists of unloading, receiving, and putting away the items intended for Company X. These items will remain in stock until a customer order requires fulfillment. when such an order is placed, the order will be picked by an order picker, packed, and delivered with the use of trucks to the customer. There is a possibility that the customer is not satisfied with an order and in this case, the order can be sent back to the Company X Returns Center. Figure 1.1 shows an overview of the logistical process described in this paragraph.

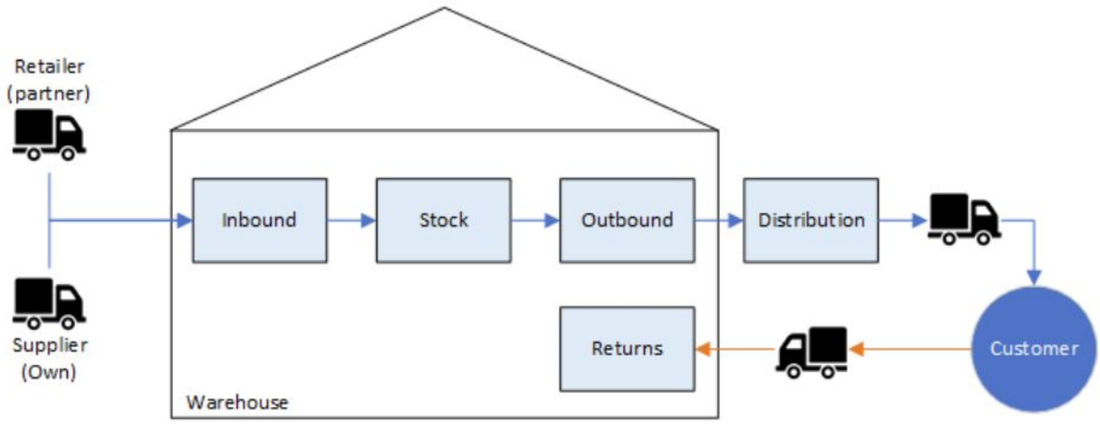

Figure 1.1: Overview of the logistical process

1.3 Problem statement

The motivation for this research stems from frequent observations of a significant number of trucks arriving at the warehouse carrying a relatively low quantity of pallets designated for Company X. After conversing with coworkers from various logistical departments, it became evident that the issue was not related to the fill rate of incoming trucks. Instead, the problem appears to arise from an uneven distribution of daily truck arrivals, which leads to substantial arrival peaks in the morning. As a consequence, the volume of items entering the warehouse experiences large fluctuations, adversely impacting subsequent stages of the inbound process. To gain a more comprehensive understanding of this problem, the problem cluster depicted in Figure 1.2 is created.

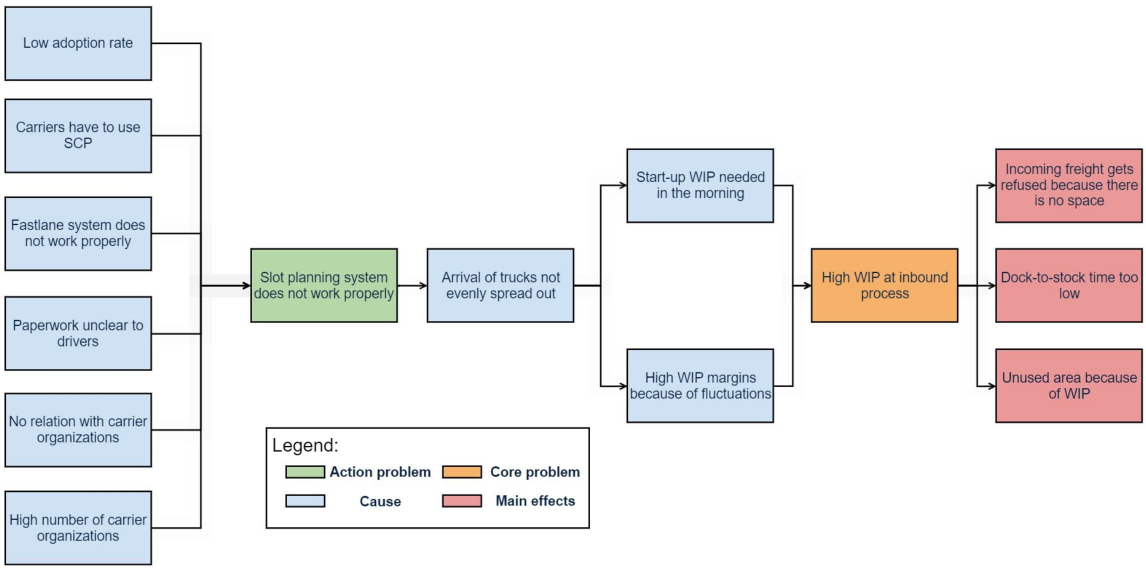

Figure 1.2: Problem cluster

When the receiving operators start working at 7 a.m., there are insufficient pallets arriving at the warehouse to occupy them for the initial two hours. To prevent this process from becoming idle, a number of pallets from the previous day is retained, enabling operators to continue working until additional pallets arrive at the warehouse. These pallets waiting to be received are called the Work In Progress (*WIP*) in this research and are located in between the unloading - and receiving process (as illustrated in Figure 1.3).

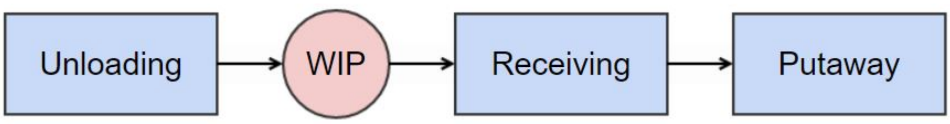

Figure 1.3: WIP location

The pallets located in the WIP area occupy space on the work floor and because the space is limited, each warehouse has a WIP limit. Monitoring WIP levels at each warehouse is important for safety - and productivity reasons. Incoming workload volumes fluctuate significantly throughout the day, so WIP margins are maintained at higher levels than desired by the inbound operation to accommodate these differences. Daily WIP margins are determined based on the production plan for that specific day and can be adjusted if the plan changes.

The relatively high volume of WIP causes three main problems for the inbound operation. Firstly, WIP consumes additional space, reducing the room available for additional pallets. Up to a certain amount of WIP this is not an issue, but there have been occasions in the past where shipments had to be refused due to a lack of space. Secondly, the over usage of space also causes those spots cannot be used for activities that have a positive contribution to the inbound process. Lastly, a high WIP has a negative impact on the dock-to-stock time. This is the total amount of time that it takes for an item to be docked and put into the stock of the warehouse. Although Company X aspires to attain a dock-to-stock time of 72 hours or less, this objective is not always achievable due to the WIP.

Carrier organizations make their own decisions concerning the moment of arrival at the warehouse, leading to morning peaks and fluctuations throughout the day. To mitigate these fluctuations, scheduling truck arrivals at different timeslots throughout the day seems like a logical solution. Company X offers carriers the option to book their own timeslot in advance, in exchange for a faster throughput at the gate of the warehouse. Based on the number of goods from the carrier, an estimate is made of the amount of time it takes to unload this shipment. Subsequently, the system checks which docks are available for this amount of time, and the transporter can reserve one of the available time slots. However, the system has not been as successful as intended. One reason is that carriers must use the Supply Chain Information System, a third-party Software as a Service (*SaaS*) that registers and processes all shipments, but which carriers found difficult to use. Additionally, the idea of prioritizing trucks at the gate did not always work, and drivers still had to wait in line. Since suppliers or partners handle transports via their own networks, Company X has difficulty making clear agreements regarding the use of the slot planning system and necessary paperwork. Furthermore, the large number of transport companies involved only adds to the complexity of the problem.

To summarize, the action problem arising from the problem orientation is the lack of a well-functioning way of scheduling the daily truck arrivals so that they are evenly spread out over the day.

1.4 Scope

To prevent the research area from becoming too broad and complex and considering the limited amount of time available for this study, limitations, and boundaries have been set.

Company X has in total 6 different warehouses, but this numerical study only focuses on Warehouse 1. This warehouse is responsible for approximately 40% of the total incoming volumes of small to medium-sized items. However, a certain strategy could be implemented in the other warehouses too.

The research only considers incoming pallets and not incoming parcels, which are usually processed immediately and thus have a negligible impact on WIP calculations.

This study focuses on the total incoming flow of pallets from Company X's own retailer shipments. The total WIP at the inbound phase does not make a separation between those two flows so both end up in the same WIP.

1.5 Research questions & research design

The aim of this study is to develop a supplier timeslot strategy that promotes a consistent flow of items into the warehouse over the course of the day. The expected outcome is a reduction in the amount of WIP at Warehouse 1 which is located between the unloading - and receiving processes. The main research question that stems from this goal and anchors this research is the following:

"How can long-term inbound shipment planning be used to reduce the amount of daily WIP at the inbound area of the Company X's Warehouse 1?"

In order to address the research inquiry and achieve the objective of this study, the main question has been divided into a number of sub-questions. Each of these sub-questions forms a component of the research methodology and will be treated in a separate chapter.

Chapter 2. Context Analysis: What does the current problem context look like?

This chapter will analyze the current state of the truck arrival process, the daily amount of incoming workload, and which Key Performance Indicators (*KPIs*) should be considered to measure the process. It is divided into the following questions to be answered:

2.1 What does the operational inbound process look like?

2.2 How does the daily truck arrival graphically look at the warehouse?

2.3 Which KPIs should be considered to measure the process?

2.4 How do the KPIs perform in the current situation?

Chapter 3. Literature Research: What relevant knowledge from the literature is available on the subject of inbound truck scheduling in the context of our research problem?

The next chapter consists of a literature review in which the following questions will be addressed:

3.1 Which important characteristics are associated with scheduling?

3.2 How can scheduling optimization problems be solved?

Chapter 4. Model Design: How should the model for the inbound truck scheduling problem be designed?

After the current situation is analyzed and the literature is consulted for the necessary knowledge, the problem-solving model will be designed. The design phase progresses according to the following design questions:

4.1 What should the problem-solving model be capable of?

4.2 What does the mathematical problem formulation look like?

4.3 How should the problem-solving algorithm be constructed?

Chapter 5. Experiment Design: How should the experimental phase be designed?

This chapter focuses on designing the experiments and tuning the parameters for our algorithm. The following questions will be central in this chapter:

5.1 Which algorithm tuning settings should be used for performing the experiments?

5.2 How can the Simulated Annealing algorithm be validated?

5.3 Which experiments should be conducted?

Chapter 6. Experimental Results: How does our algorithm perform?

Chapter six presents the analysis of the results obtained from the conducted experiments. These results provide insights and answers to the following questions:

6.1 How does the algorithm perform with the baseline settings?

6.2 How can the lessons learned be translated into practical implications?

6.3 How valid are the results of the algorithm?

6.4 What is the performance score of the algorithm compared to the current and optimal situation?

Chapter 7. Model Implementation: How should the timeslot indication model be implemented in practice?

The seventh chapter considers the implementation of the designed schedule optimization tool. The experiences of the failed implementation of the old scheduling tool play an important role in the design of the implementation phase. These practical facets will be covered by the next two questions:

6.1 What do the different implementation phases look like?

6.2 How can the lessons learned be translated into practical implications?

The schematic version of the different chapters linked together, and thus the research methodology, is visualized in Figure 1.4. Both chapters two and three will be used as input for the design of the model (chapter four). After the design phase, the experiment design phase starts and the experimental results will be discussed in this phase. The final phase is the implementation, which explains the various stages of implementation.

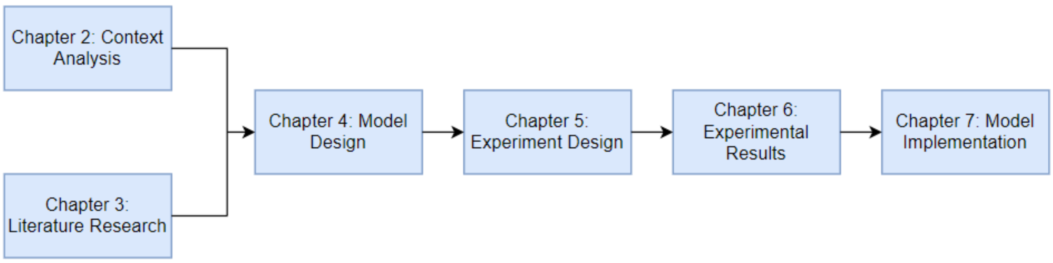

Figure 1.4: Research methodology

This report ends with a conclusion that will answer the main question and with recommendations for the company Company X concerning the impact of the scheduling model and its implementation.

Chapter 2

Context Analysis

This chapter describes the context analysis of the research problem. Section 2.1 describes the different processes of the inbound operation at Warehouse 1. Section 2.2 shows the analysis of the daily truck arrivals in terms of shipments and the amount of workload each truck carries. Section 2.3 elaborates on the productivity numbers of the unloading - and receiving process. Section 2.4 presents the On-Time Performance and section 2.5 addresses the way the WIP is calculated. The final section 2.6 will answer the sub-question "What does the current problem context look like?".

2.1 The inbound process

Each incoming shipment undergoes several sub-processes before it is stored in inventory. This subsection provides a simplified version of the inbound process, which includes: truck arrival, unloading, receiving, and putaway.

2.1.1 Truck arrival

The inbound process starts with the arrival of trucks (Figure 2.1) carrying the items designated for Company X. Some of the carrier organizations have scheduled themselves a time slot, but as section 1.3 describes, the majority of them arrive at their own convenience. Before entering the warehouse premises, they are required to present necessary documentation at the gate, certifying that they are delivering a shipment for Company X. There are three lanes in front of the gate, two for regular deliveries and one for fastlane delivery. Carriers with a reserved time slot can utilize the fastlane by only showing their appointment number, provided that they arrived on time. Upon completing the formalities at the gate, the driver is assigned to a dock where the handling of the truck begins.

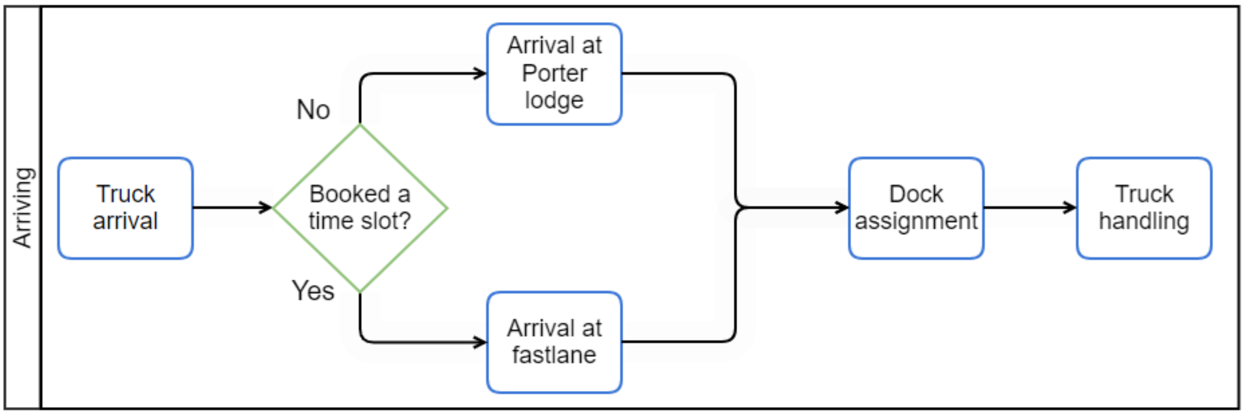

Figure 2.1: Arrival process

2.1.2 Unloading

Once the truck is parked at the assigned dock, the unloading process starts (Figure 2.2). Subsequently, the pallets will be unloaded from the truck by the truck driver himself under the guidance of an operator. Then the number of pallets is counted and the delivery conditions are verified. Finally, after signing the appropriate paperwork, the shipment is scanned on load carrier level and any violations of the delivery terms and conditions are registered. The scanned load carriers are registered in the Warehouse Management System (*WMS*) to maintain an overview of all unloaded shipments.

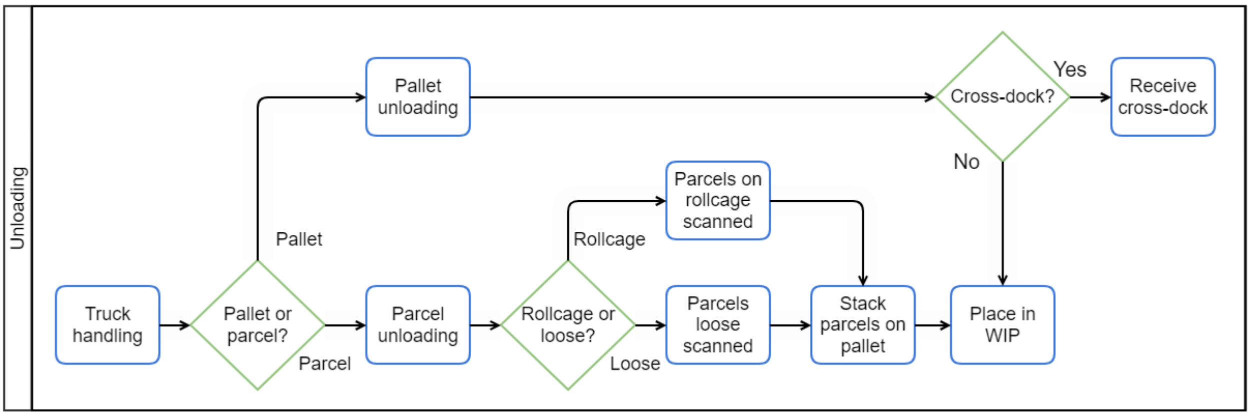

Figure 2.2: Unloading process

The unloading of parcels consists of additional process steps, as they arrive on roll containers or loose. All parcels are extracted from these containers and consolidated onto pallets along with the loose parcels. Afterward, all pallets are forwarded to the receiving process or placed in the WIP area. An exception is made for regular pallets that are intended for cross-dock shipments because those will be received in the cross-dock area.

2.1.3 Receiving & Putaway

The first step in the receiving process (Figure 2.3) involves placing the unloaded pallets in the WIP area, provided that there is insufficient space in the receiving area. As section 1.4 describes, most of the pallets filled with parcels are directly transferred to the receiving area and thus almost never end up as WIP. Therefore, they are not further considered in this study. When a pallet is next in line, it undergoes decantation. The process of decanting involves unloading items from the pallets and placing them into blue bins referred to as 'totes'. Before an item is put into a tote, it is scanned and marked as 'received' in the WMS. It happens that the item is not yet registered in the system (new) or requires an exception to be made before it can be placed in the tote. In such cases, the item is sent to the exception area where it is registered before allocating it to the tote. Once a tote is filled with items, it is transported via an automatic roller conveyor through the warehouse, where the putaway process starts.

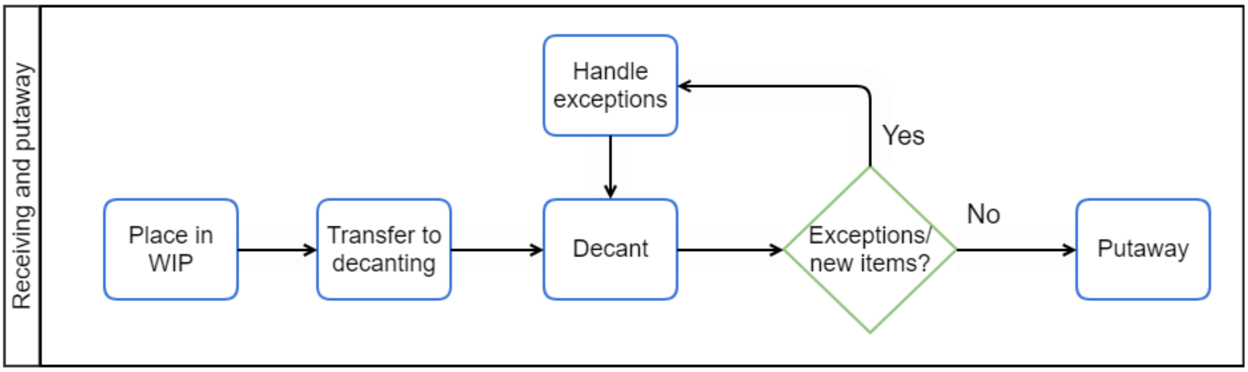

Figure 2.3: Receiving process

The final stage of the inbound process involves stocking the items in the warehouse, a process referred to as "putaway" (Figure 2.3). Once an item has been put away, it is scanned, and the time elapsed from unloading until putaway is calculated. Company X guarantees its partners that incoming goods will be stocked within 72 hours from the moment of unloading.

2.2 Arrival analysis

The arrival of the trucks can be divided into two parts, namely the timing of arrival and the amount of workload being delivered. This section describes the analyses conducted on both components.

2.2.1 Truck arrivals

The observation by the inbound department is that the daily truck arrivals are not evenly spread out over the day which causes arrival peaks in the morning. To confirm this statement and to gain a clear understanding of the actual truck arrival times, we analyzed all incoming shipments from March 3rd until October 28th of the year 2022. This period was chosen because data before this time was unreliable due to the data retention policy.

Each shipment receives a unique 'ShipmentID' when it is booked in the Supply Chain Information System, and this ID is used to track its arrival time at Warehouse 1 once it has been unloaded and scanned. However, it should be noted that a single ShipmentID could consist of multiple truck arrivals throughout the day. Unfortunately, the available data do not provide information about the number of trucks in a shipment and their arrival times, but only about the first moment of arrival. Therefore, for the purpose of the analysis, we assume that all shipments arrive in one truck at the registered arrival time. To create a histogram of the arrival times, we extracted all ShipmentIDs and their corresponding arrival times from the Supply Chain Information System database and used a bin size of 15 minutes between 7:00 a.m. and 11:59 p.m. The unloading process officially ends at 6 p.m., although exceptions are occasionally made for trucks arriving after this time. Figure 2.4 presents the resulting histogram:

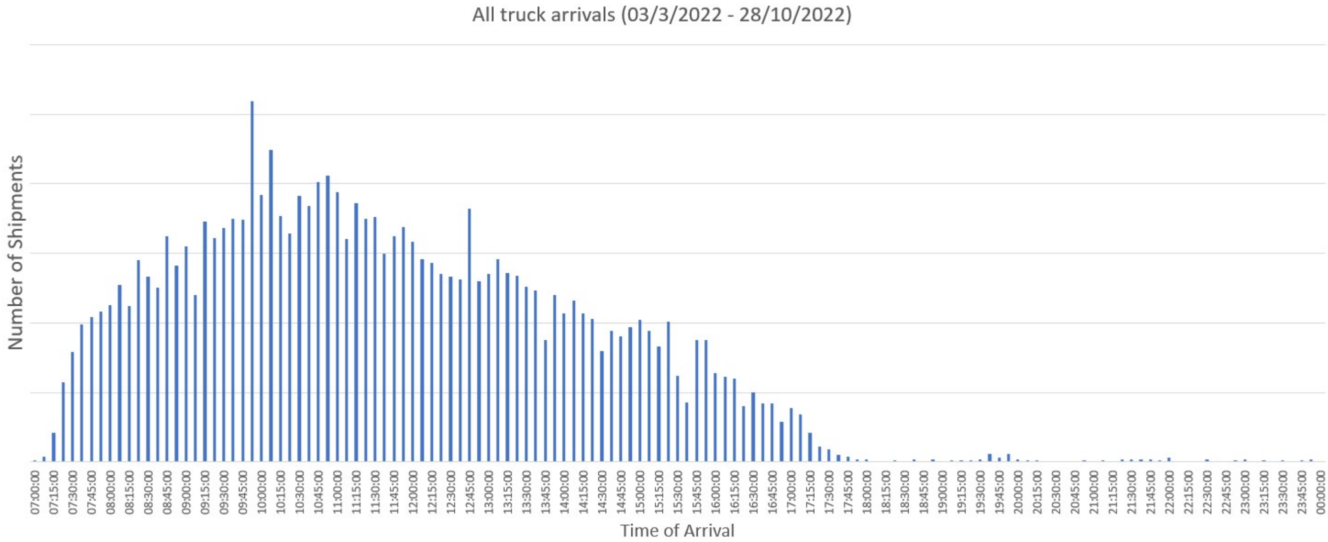

Figure 2.4: Truck arrival times

The first thing to notice about the histogram in Figure 2.4 is that there are more shipments (65%) coming in before 12:30 p.m. than after. Visually, it seems that the distribution is skewed to the right and that the truck arrivals peak in the morning, which is consistent with the perception of the inbound department. We made the same histograms for each weekday to make sure that there is not a weekday which is an exception to this skewed distribution. An overview of these histograms is included in Appendix A. These additional histograms also visually show that there is peak formation in the morning which declines during the afternoon. We analyzed the summary statistics of the histogram of all truck arrivals (Figure 2.4) to confirm this visual conclusion.

Table 2.1: Summary statistics all truck arrivals

Summary Statistics all shipments	
Mean	11:34:56
Median	11:15:32
Skewness	0.5151258
Minimum	06:58:06
Maximum	23:49:26
Count	10016

Table 2.1 shows that the skewness of the distribution of all truck arrivals is rounded up to 0.52. When the skewness value is larger than 1 or lower than -1, the distribution is strongly skewed to the right or left, respectively. A moderately skewed value is one between 0.5 and 1 or -0.5 and -1. The distribution is considered to be fairly symmetrical if the value occurs to be between -0.5 and 0.5. From the skewness value of 0.52, it can be concluded that the distribution is slightly skewed to the right and this is consistent with our visual experience with the histogram.

Also, the skewness value of the individual weekdays is all between 0.4 and 0.6 (see Appendix A), confirming that the daily truck arrival is not evenly distributed and has peak formation in the morning.

2.2.2 Incoming workload

Besides the arrival time of trucks at Warehouse 1, we also analyze the number of pallets delivered. By combining 2 different tables from the Supply Chain Information System it is possible to determine the number of load carriers assigned to each ShipmentID. Each load carrier, in this case a pallet, has a unique load carrier label, so when a ShipmentID has 4 different load carrier labels the shipment consists of 4 pallets. Figure 2.5 shows the number of pallets per shipment.

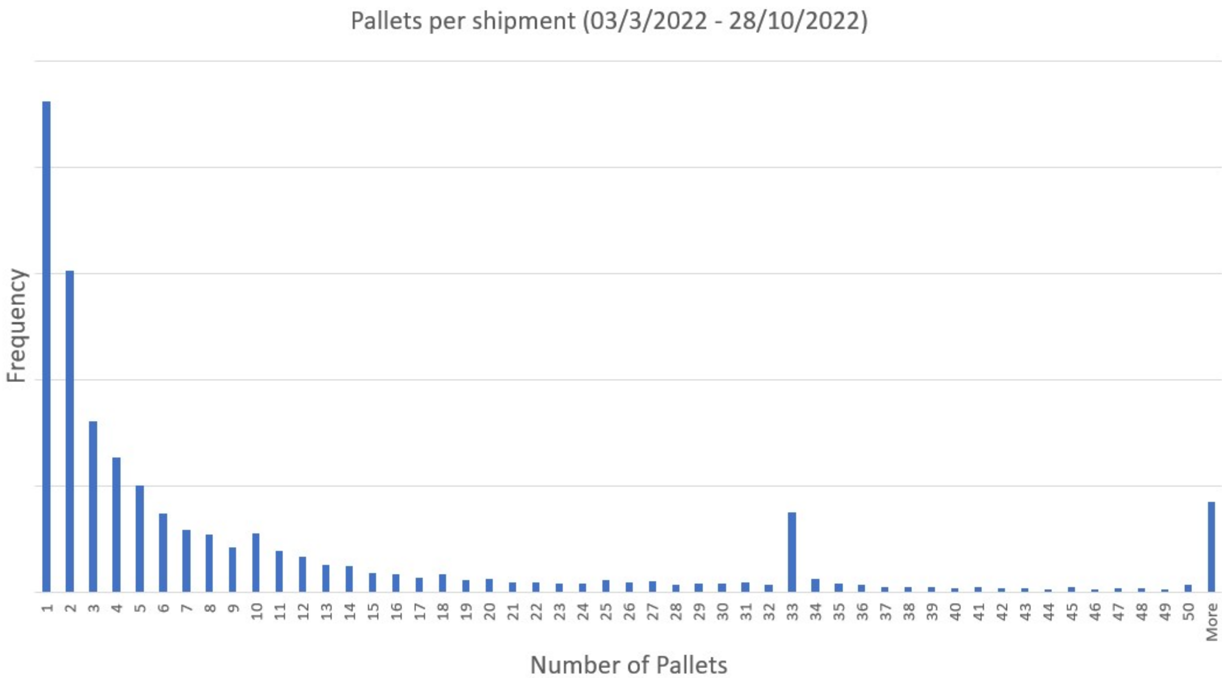

Figure 2.5: Amount of pallets per shipment

The data analysis reveals a reduction in the frequency of shipments as the number of pallets per shipment increases, with a notable spike occurring at 33 pallets. This can be explained by the fact that an average truck can carry a maximum of 33 euro pallets. One of the partners has contractually negotiated a purchase discount with some of its suppliers that apply to each full truck they deliver.

2.3 Productivity

Productivity is the most important KPI within inbound operations, as it reflects process speed and operational costs. This metric is calculated as the number of actions completed per person per hour, where the definition of action varies depending on the process step. For instance, in the unloading process, an action is considered as scanning a pallet after completing the steps described in subsection 2.1.2. Similarly, scanning an item before placing it in a tote represents the corresponding action in the receiving process. Within the inbound operation, they particularly pay attention to the productivity of the receiving process, which involves the most significant amount of operational hours and associated costs. We observe both unloading - and receiving productivity to obtain a clear indication of the processes preceding and following the WIP area.

2.3.1 Productivity unloading

Within the process of unloading, productivity is assessed based on the number of handling devices (*HDs*) scanned per person per hour. The HDs can include both pallets and loose parcels. The scanning itself is a simple and quick operation, but as subsection 2.1.2 describes, it cannot be performed until all previous steps are completed which is time-consuming. The unloading productivity numbers for the period of weeks 39 through 45 of FY 2022 are presented in Figure 2.6:

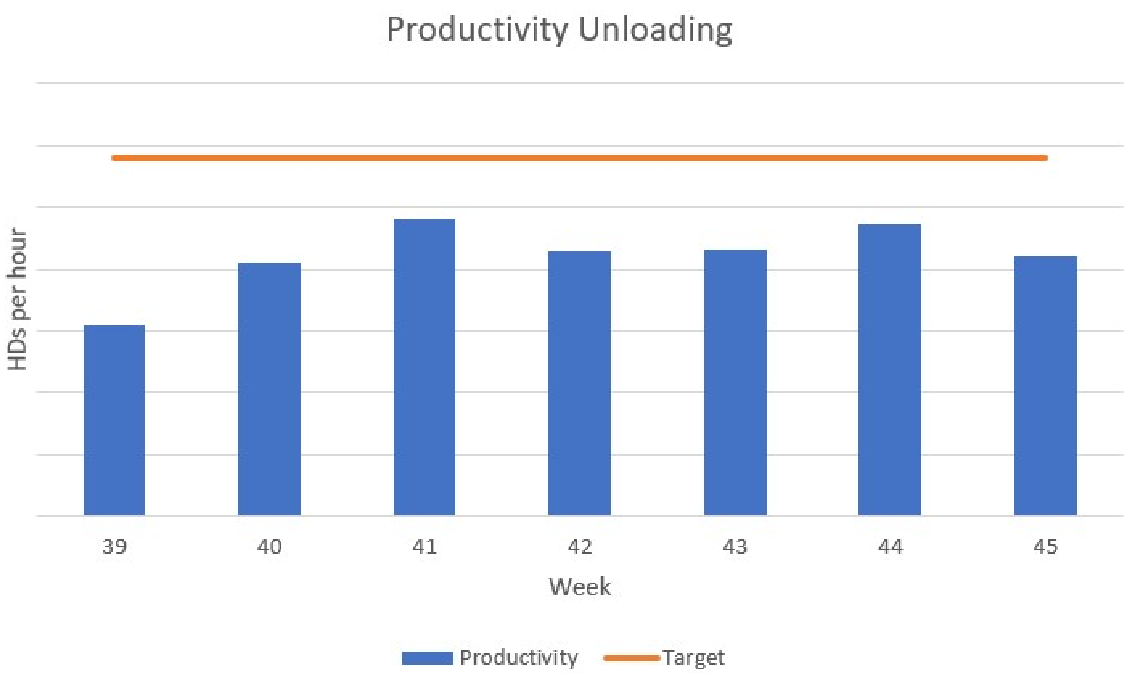

Figure 2.6: Productivity unloading weeks 39-45 (the year 2022)

Figure 2.6 shows that the weekly productivity goal set by the inbound department is approximately 29 handling devices per hour. However, the graph shows that this target was not achieved during any of the production weeks, with the lowest productivity level of 15.5 HDs per hour observed in week 39. The primary reason for this shortfall is the mismatch between the number of personnel scheduled to work and the incoming amount of workload.

2.3.2 Productivity receiving

The productivity of receiving is expressed as the number of items scanned per person per hour before it is placed in a tote. Repeatedly, we analyzed the productivity numbers for weeks 39 through 45 in the following figure:

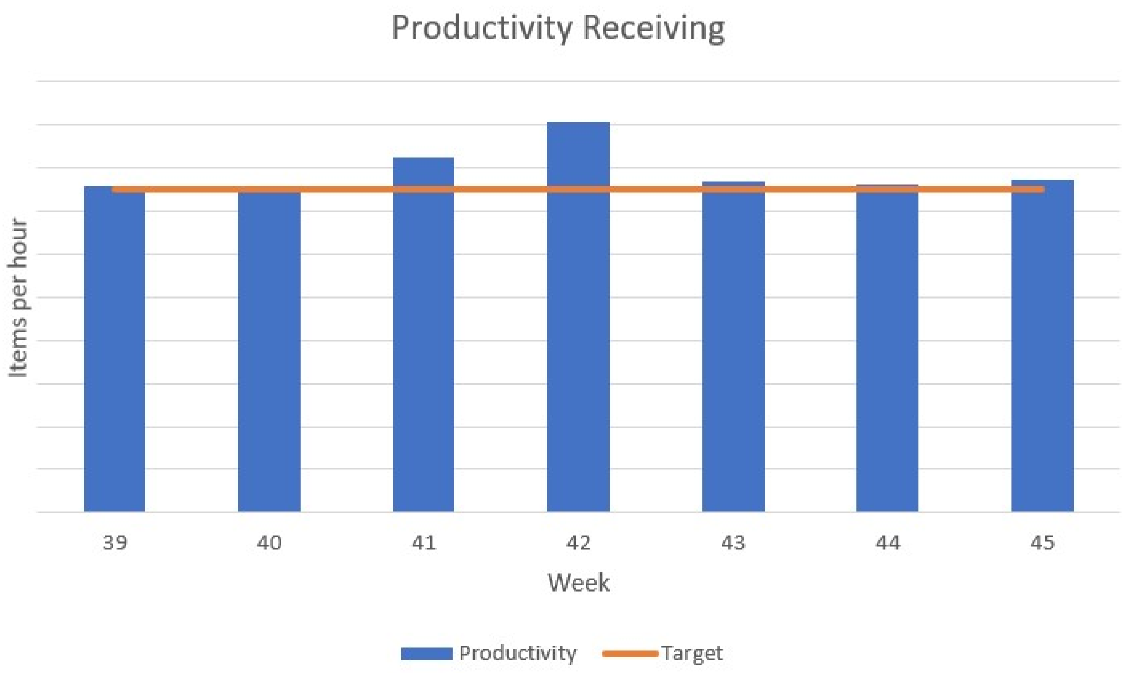

Figure 2.7: Productivity receiving weeks 39-45 (the year 2022)

As depicted in Figure 2.7, the productivity targets for the receiving process were reached in nearly all weeks. Only the productivity of week 39 is slightly below its target. After week 39, the target was reset to a new value, which was attained in the subsequent weeks. Instead of the productivity of the unloading, the productivity target of the receiving phase is attained in almost all weeks. This is because the receiving area consists of a number of workstations in which the input of incoming work is more constant than the unloading phase.

2.4 On-Time Performance

Another important KPI for the inbound organization is 'On-Time Performance' (*OTP*). This KPI measures the percentage of items that have been put into stock within 72 hours from the moment of unloading. The WIP has a major influence on the percentage of items that are put into stock within the agreed time, so this indicator is interesting to take into account during this study. As the WIP increases, the duration of items in the WIP area also increases, leading to a higher dock-to-stock time. Figure 2.8 presents a dashboard that indicates the OTP within Warehouse 1.

Figure 2.8: Warehouse 1 On-Time Performance 2022

The horizontal axis in Figure 2.8 shows the different week numbers starting from the first week of January and the vertical axis represents the percentage of items stocked within 72 hours of unloading. The graph consists of three lines with different colors. The red line ("Processing Time Available") indicates the percentage of the incoming items from which data is available. The higher this value is, the more reliable certain statements can be made regarding the OTP. The line in blue indicates the actual value of the OTP and the green line (added by ourselves) is the target of 83% they maintain. This target is based on experiences from practice and no mathematical calculations have been performed for getting this percentage.

It is clearly noticeable that the available data per item has grown rapidly over the past year and currently more than 90 percent of the item data is available. The percentage of items stocked within 72 hours shows significant fluctuations with the result that the target is usually not met. When the value falls below the target, certain actions can be taken such as prioritizing shipments that are in danger of being late or increasing the application of First Come First Served (*FCFS*). However, this target value is not considered a rigid threshold, so the aforementioned actions are not always taken.

2.5 Work in Progress

Currently, a certain level of WIP is essential for the inbound operations to ensure high productivity in the processes of unloading and receiving. However, the amount should not be too high and in an ideal world, processes can be executed without having a WIP. As subsection 2.1.3 describes, the pallets packed with units waiting to be processed and seen as WIP are located in between the unloading - and receiving phases. To calculate the daily number of units present in the WIP area, we analyze the number of units unloaded and received. The difference between these two values plus the previous day's WIP yields the current day's WIP. It is important to note that these calculations are only done at the end of the day, so there are no WIP values available for other moments in time. This means that the calculated WIP values are an optimistic version of reality. In fact, there is a big chance that during the day the WIP is higher than the calculated number, since more units come in during the morning hours than can be processed, leading to an increase in the WIP. As the incoming units decrease throughout the afternoon, the WIP reduces until it reaches the calculated end-of-day value.

After the level of WIP is calculated, it is determined if the calculated value is acceptable or if corrective measures need to be taken when the level exceeds certain boundaries. These boundaries are established based on the production plan for the corresponding day. The production plan is the number of units expected to arrive on a particular day. If the production plan indicates the arrival of fewer than 100% units, then the ideal WIP value is at 33% units with a minimum and maximum of 25% - and 42% units, respectively. A plan with more than 100% units corresponds with an ideal WIP value of 50% units and a limit of a minimum of 42% - and a maximum of 59% units.

To regulate the WIP level, the inbound department uses a model that allows a deviation of up to 8% units from the ideal WIP value. In a 'WIP meeting', the team decides which measures should be taken to move the WIP value towards the target if the deviation is within these limits. If the WIP value is outside these margins for more than 3 days, then concrete actions will be taken and those can be the following:

- Upscaling/downscaling of the inbound operations;
- Increasing/decreasing the order volume;
- Cancel suppliers if the order is already placed (if WIP is too high).

It is meaningful to remark that the values mentioned above have arisen from practical experience and that there are no mathematical calculations done to get those values. Therefore, terms such as "ideal", "minimum" and "maximum" do need to be placed in the context of the current situation which is not optimal by itself.

To understand how the WIP values including the limits looked like over the past few months, we used the production plans and the calculated WIP stands to incorporate them into line charts. Figure 2.9 and Figure 2.10 are examples of line charts of weeks 42 and 45, respectively, and these clearly show the impact of the production plan on the allowed limits:

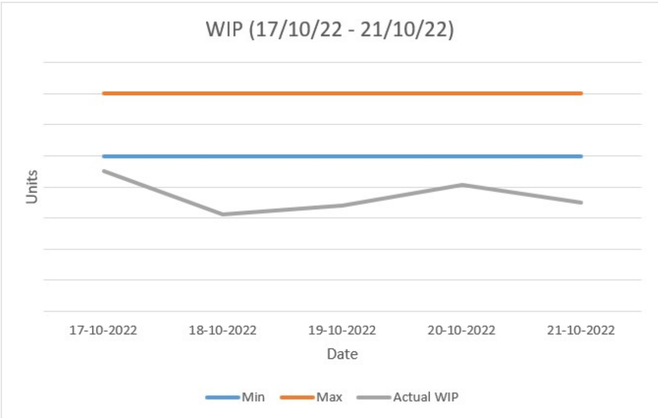

Figure 2.9: WIP week 42

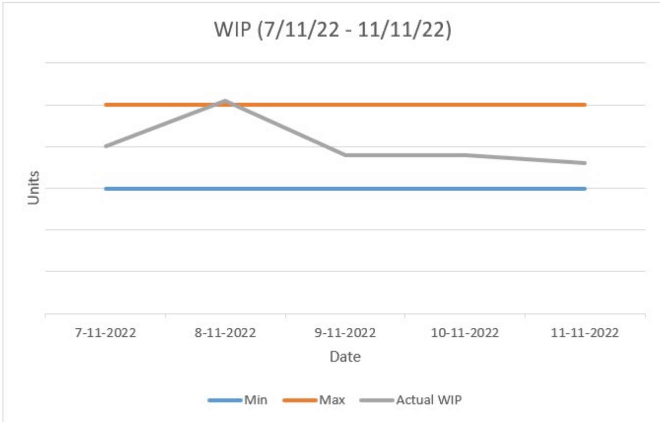

Figure 2.10: WIP week 45

In week 42 (Figure 2.9), the production plan exceeded 100% units every day, resulting in limits of 42% - and 59% units. It is evident that the WIP position was too low each day, but not to the extent that rigorous actions were taken to increase the WIP. Similarly, Figure 2.10 presents the WIP numbers for week 45, which were also roughly between 25% - and 42% units. The big difference compared to week 42 is that the daily production plans were all below 100% units, so the actual WIP remained within the boundaries. In the last three days, the WIP approached the ideal value of 33% units, potentially due to actions taken on Tuesday.

The level of WIP is not only affected by the uneven arrival of trucks throughout the day but also by other factors such as:

- Unpredictable inbound flow: large deviations in not only the quantity but also the type of load carriers can disrupt the inbound operations;
- Deviation of available capacity: capacity may get lost due to machine failure;
- Incorrect forecast model: there have been days in the past where the method of forecasting was incorrect, which resulted in more units (e.g. +25% units) coming in than expected;

- Inexperienced or insufficient workforce: the number of operators scheduled is based on the expected production plan. If reality deviates, there may be too few operators present which can increase the WIP. It is also possible that there are too many inexperienced operators on the job, which could slow down the speed of an operation and increase the WIP.

2.6 Conclusion

To address the sub-question of "What does the current problem context look like?", we mapped the current inbound process and analyzed the daily truck flow. It is evident that the daily arrival ratio is skewed, with significantly more trucks arriving in the morning than in the afternoon. Furthermore, due to insufficient incoming volume, the unloading productivity falls way below the weekly targets. Finally, there are significant fluctuations in the weekly on-time performance and the performance of the "Work In Progress" KPI. These are sufficient reasons to conclude that the current inbound process is underperforming in multiple aspects and that a solution must be sought to reduce the amount of work in progress.

Chapter 3

Literature research

This chapter answers the following question with the use of a literature review: "What relevant knowledge from the literature is available on the subject of inbound truck scheduling in the context of our research problem?". The first section 3.1 explains the fundamental concept of scheduling and the characteristics of our inbound truck scheduling problem. Section 3.2 elaborates on inbound truck scheduling and conducted research on this subject. Section 3.3 explains the concept of metaheuristics and the use of Simulated Annealing for our research. The final section 3.4 provides a summary of the key takeaways from the literature review and answers the sub-question.

3.1 Scheduling

The process of scheduling is used in many manufacturing and services industries to assign resources to tasks over given time periods with the aim of optimizing one or more different objectives [13]. These tasks, also called jobs, and resources can take a variety of shapes, and in our research, the resources are represented by the docks at the warehouse. The jobs are the incoming trucks filled with pallets destined for Company X whose workload is based on the number of pallets (the variable factor) plus the standard time it takes to (un)dock the truck (the constant factor).

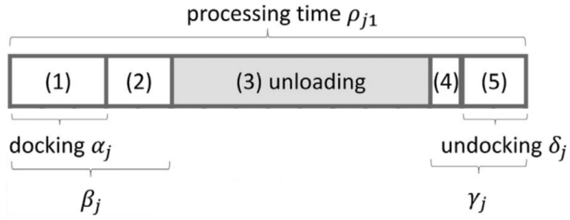

Figure 3.1: Processing time unloading

Figure 3.1 is a schematic example from an article from Tadumadze et al. [15] and it represents the total time it takes to complete one unloading job. The beta value represents the total processing time of all the steps required before the unloading can start (including docking) and the gamma value represents the actions after unloading (including undocking).

Eventually, all individual jobs will be scheduled over the available machines (docks), and an overview of which jobs are assigned to which docks can be displayed in a Gantt chart. An example of such a schematic overview is given in Figure 3.2:

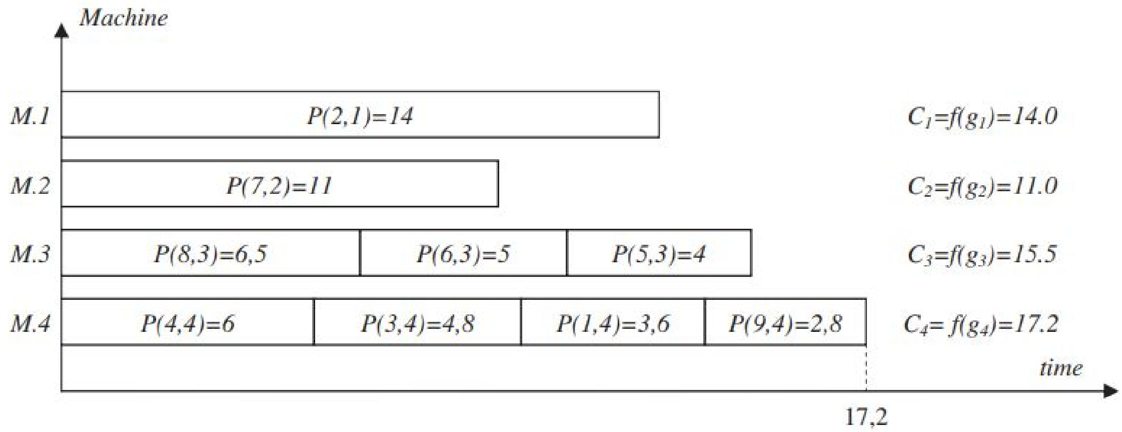

Figure 3.2: Gantt chart example

Figure 3.2 consists of two different axes. The x-axis represents the total processing time each individual machine took to perform the jobs (the longest is 17.2 minutes) and the vertical axis stands for the different machines. This example origins from an article written by Balin [1] and he used a genetic algorithm to schedule independent jobs on non-identical machines in order to minimize the total makespan.

3.1.1 Identical parallel machine scheduling problem

The warehouse our research focuses on exists of a number of parallel docks which all have the same processing time per incoming truck, depending on the number of pallets the truck is carrying. Whenever a job can be processed on any one of the parallel machines and all the machines are identical, one speaks of an identical parallel machine scheduling problem [13]. An example formulation of such a mathematical model is given by Mokotoff [12]:

Indices

i	index of jobs
j	index of machines

Parameters

p_i	processing time of job i
N	total number of jobs
M	total number of machines

Decision variables

y	makespan

$$x_{ij} = \begin{cases} 1, & \text{if job i is assigned to machine j} \\ 0, & \text{otherwise} \end{cases}$$

Objective function

$$\text{min y}$$

Constraints

$$\text{s.t.} \quad \sum_{j=1}^{m} x_{ij} = 1, \quad 1 \leq i \leq n, \tag{3.1}$$

$$y - \sum_{i=1}^{n} p_i x_{ij} \geq 0, \quad 1 \leq j \leq m, \tag{3.2}$$

$$x_{ij} \in \{0, 1\}, \quad 1 \leq i \leq n, \quad 1 \leq j \leq m \tag{3.3}$$

The objective of this particular example is to minimize the makespan, which is the total time it takes to finish all the jobs. Constraint (3.1) ensures that each job is processed only once and constraint (3.2) calculates the makespan per dock. The last equation (3.3) forces the binary variable x_{ij} to have a value of 0 or 1.

Our model will have an objective function that aims to ensure a constant flow of items going into the warehouse. The principle of a binary variable indicating whether a job is assigned to a machine will be utilized in our model.

3.1.2 Offline vs. online scheduling

Within scheduling a distinction is made between offline - and online scheduling. The main difference between these two variants is the extent to which the information about the jobs to be scheduled is known in advance. This information contains the number of jobs, the processing times, the weights, the release dates, and the due dates. In the case of an offline schedule, the decision-maker can create the entire schedule before the processes start since all the required data is known [13]. In the case of an online scheduling problem, this information is not (completely) known beforehand and the decision-maker will find out about it when a job is released. When this information becomes known when a job is released, it is called a *clairvoyant* online scheduling problem. The opposite of clairvoyant is *non-clairvoyant* and within these cases, the information becomes known after the job is completed. The input of our model consists of complete known historical data, which means that we engage in offline scheduling.

3.1.3 Deterministic vs. stochastic models

There exist two types of areas within the field of operations research: deterministic operations research and stochastic operations research. Within a deterministic model, all parameters are fixed, while the parameters in a stochastic model could be random [8]. For example, in the case of a scheduling model, the job processing times would be random, or the moment a job releases is randomly distributed. This may lead to uncertainty within an optimization problem and Chaari et al. [6] define uncertainty as follows: "Uncertainty is related to doubts concerning the validity of knowledge or to not knowing if the proposition is true or not". It is important to keep in mind that in real life there is a difference between the preannounced - and the actual inbound shipments arriving at the warehouses of Company X. However, for the sake of simplicity, we exclude that consideration from our model, rendering our optimization model deterministic.

3.1.4 Time intervals

When formulating a mathematical model, it is possible to incorporate time in two different ways, either discrete or continuous. In the former, the time horizon is divided into a set of uniform intervals, which are taken into consideration in the decisions made by the model. Starting a task can only occur within a specific time interval. A greater number of intervals leads to a more accurate representation of reality but also increases the computational difficulty, which may result in a longer time required for the model to find an optimal solution [7].

In contrast, a continuous-time representation utilizes variable time intervals. In this case, events are linked to continuous variables that can assume any value on the time horizon, allowing for the starting of a task at any point in time. This has the advantage of providing a more realistic representation of time in the model. Both forms are depicted schematically in Figure 3.3.

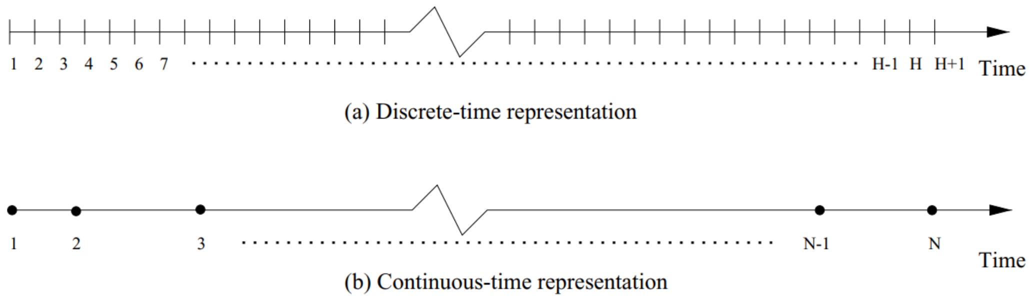

Figure 3.3: Representations of time

The goal of our research is to make the number of items entering the warehouse per time interval as constant as possible by assigning time slots to a part of the suppliers. Therefore, it is a logical choice to integrate a discrete-time representation into our mathematical model, as it makes it easier to calculate the number of items per time interval and the impact of assigning fixed timeslots to suppliers.

The integration of time intervals in the MILP model can be achieved by adding an additional index to the decision variable. An example of this is provided by Wolff et al., who have also formulated a MILP model for a truck scheduling problem [16]. They utilize the binary decision variable x_{idt}, which takes a value of 1 if truck i is assigned to door d and processing starts in time interval t.

3.2 Inbound truck scheduling

Extensive research has been conducted in the literature regarding the scheduling of inbound trucks, often within the context of cross-docking operations. Cross-docking is a logistical technique where goods are unloaded from incoming trucks, sorted, dispatched, and promptly reloaded onto outbound trucks [10]. Consequently, this approach maintains a relatively low inventory level within a warehouse, as the goods typically spend no more than 24 hours on-site. For our identical parallel machine scheduling problem, our focus will be exclusively on the inbound trucks segment of cross-docking.

3.2.1 Cross-docking research

Yu et al. [18] investigated a cross-docking system incorporating a temporary storage zone preceding the shipping dock. Their study aimed to identify the optimal sequence for truck docking, encompassing both inbound and outbound vehicles, with the goal of minimizing total operational duration or maximizing cross-docking system throughput. Although this approach proves viable for addressing problems of smaller instances, its efficacy diminishes and becomes unfeasible when confronted with medium to large-scale instances due to increasing computational time demands. To enhance solution efficiency, the researchers developed a heuristic algorithm.

Another study conducted within the same research domain is the work of Madani-Isfahani et al. [11]. They formulated a MILP model for a truck scheduling problem within a cross-docking system, with the primary objective of minimizing the makespan. To address this NP-hard problem, the researchers implemented two distinct metaheuristic techniques: simulated annealing and firefly algorithms.

Yazdani et al. [17] investigated a cross-docking challenge involving multiple inbound doors, as opposed to the more prevalent research focus on cross-docking problems with a single inbound door. Smaller instances of this problem were optimally solved using the CPLEX solver, while for larger instances, the researchers developed a metaheuristic approach incorporating, among other methods, Simulated Annealing.

The literature review conducted by Buakum and Wisittipanich [5] regarding metaheuristic solutions proposed for cross-docking operational problems within the period from 2001 to 2017, underscores the significance of utilizing metaheuristics. Given that these problems are classified as NP-hard, exact algorithms prove effective only for smaller instances. Conversely, real-world instances of these problems tend to be larger and more complex, thereby rendering metaheuristics a more viable approach.

All the aforementioned studies share a common thread: when addressing inbound truck scheduling problems, exact solutions are attainable within a reasonable amount of time only for small problem instances. In the case of larger problem instances, a scenario frequently encountered in real-life situations, the application of metaheuristics becomes essential. Considering that our dataset spans multiple days and, is of considerable size, we approach the solution for our inbound truck scheduling problem using a metaheuristic.

3.3 Metaheuristics

When attempting to solve optimization problems, it is not always possible to achieve an optimal solution using an exact method when dealing with large problem instances. In such cases, heuristics are frequently used to obtain feasible solutions, particularly when the problems are large and NP-hard. However, it should be noted that using heuristics does not guarantee optimal solutions, but rather good approximations. In addition to heuristics, there also exist metaheuristics [14]. Unlike heuristics, which are problem-specific, metaheuristics are generic problem-solving frameworks that can be applied to a variety of optimization problems. Constructive and improvement heuristics are subcategories of heuristics, where constructive heuristics can be used to create an initial feasible solution, and improvement heuristics can enhance the quality of the feasible solution.

Metaheuristics are a type of heuristic that balances the trade-off between intensification and diversification to overcome local optima. A local optimum is the best solution within a small neighborhood of possible solutions. Intensification involves exploiting an area with promising solution characteristics, while diversification entails exploring the entire feasible region [14].

An example of a metaheuristic is Simulated Annealing, which has been demonstrated by Boysen et al. [4], Madani-Isfahani et al. [11] and Yazdani et al. [17] as effective in solving inbound truck scheduling problems. For this reason, we have chosen to apply Simulated Annealing as a metaheuristic to solve our inbound truck scheduling problem.

3.3.1 Simulated Annealing

Simulated annealing is a probabilistic method that is used to find a global optimum or minimum and it helps avoid getting stuck in local minima or maxima [2]. The algorithm was introduced by Kirkpatrick et al. [9] and is inspired by the annealing process, where a solid is heated above its melting point to allow the atoms to move freely and change randomly through various states. Similarly, in Simulated Annealing, the algorithm accepts a solution that is not necessarily better than the current solution as the temperature T approaches zero by multiplying the temperature with a decrease factor. The probability of accepting a solution follows the *Boltzmann distribution* [14].

Blum and Roli [3] described the Simulated Annealing on a high-level as follows:

Algorithm 1: High-level description Simulated Annealing algorithm

$s \leftarrow$ GenerateInitialSolution();
$T \leftarrow T'$;
while *termination conditions not met* **do**
    $s' \leftarrow$ PickAtRandom($N(s)$);
    if $f(s') < f(s)$ **then**
        $s \leftarrow s'$ (s' replaces s);
    end
    else
        Accept s' as new solution with probability $p(T, s', s)$;
    end
    Update(T);
end
Return s;

The above Simulated Annealing Algorithm 1 forms the foundation of the algorithm we are developing to address our identical parallel machine scheduling problem.

3.4 Conclusion

The literature review has provided an overview of the characteristics of our inbound truck scheduling problem. It can be defined as an offline identical parallel machine scheduling problem, which can be formulated as a deterministic MILP model. Implementing a discrete-time representation enables us to calculate the number of items entering the warehouse per time interval.

Furthermore, studies have demonstrated that our inbound truck scheduling problem is prevalent within the context of cross-docking operations. The primary conclusion is that we will develop a metaheuristic due to the size of our scheduling problem. Based on established evidence from research, we have chosen to utilize a Simulated Annealing algorithm to address our identical parallel machine scheduling problem.

Chapter 4

Model design

The fourth chapter of our research paper focuses on addressing the sub-question "How should the problem-solving model be designed?". The first section 4.1, discusses the model assumptions that we made. The subsequent section 4.2 provides a description of what our model should be capable of. Section 4.3 elaborates on the mathematical problem formulation. Finally, this chapter concludes with section 4.4, which explains the different phases of our Simulated Annealing algorithm, and section 4.5 answers the sub-question.

4.1 Model assumptions

Due to the complexity of the real-world problem situation, it is necessary to make a number of assumptions to simplify the model:

- We assume that all trucks arrive between 7:00 a.m. and 6:00 p.m. and a day is divided into 11 intervals of 1 hour.
- Each shipment can only arrive in one truck at one point in time during the day.
- Each truck can carry a maximum number of 66 pallets.
- The maximum number of docks that can be occupied at the same time interval is 25.
- It takes 10 minutes to dock and undock a truck plus every pallet adds another minute to the total processing time.

4.2 Model description

The objective of our model is to achieve a steady flow of incoming items to the warehouse. To accomplish this goal, we will establish agreements with a selected group of suppliers regarding the timeslots for their goods deliveries. The selection of suppliers for these agreements and the allocation of their time slots are determined with the use of a Simulated Annealing algorithm. This algorithm utilizes historical company data from Company X, consisting of pallet shipments delivered across a span of 168 days. A schematic example of a potential outcome produced by the algorithm is illustrated in the following Figure 4.1:

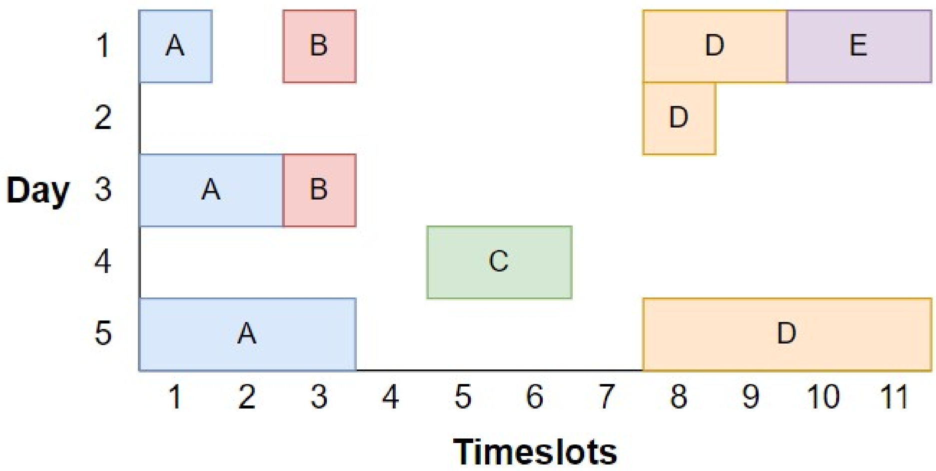

Figure 4.1: Schedule example

This example shows five different companies, denoted as A, B, C, D and E, which delivered shipments over the course of five days. Despite variations in their processing times, the same company arrives at the same timeslot. Ultimately, the Simulated Annealing algorithm will calculate the objective function, which is described in the following section 4.3.

4.3 Mathematical model

To gain a comprehensive understanding of our scheduling problem, we formulate it into a mathematical model and this section will describe and explain this model used for scheduling the arrival of trucks.

The formulation is divided into the following components: indices, parameters, decision variables, objective function, and constraints.

4.3.1 Indices, Parameters and Decision Variables

Indices

c	index of companies
d	index of days
i	index of docks
j	index of trucks
t	index of time intervals

Parameters

$DayList_j$	the designated day number for truck j
df_d	desired item flow of day d
M	an appropriately large positive number
$items_j$	items carried by truck j
$TruckToCompany_j$	the matching company of truck j
$pallets_j$	pallets carried by truck j
$proc_j$	total processing time of truck j
s	sum of items
$Trucks_c$	total number of trucks with a shipment of the company c
C	set of all companies
D	set of all days
I	set of all docks
J	set of all trucks
T	set of all time intervals

Decision variables

$$X_{ijdt} = \begin{cases} 1, & \text{if truck } j \text{ is assigned to dock } i \text{ at interval } t \text{ on day } d \\ 0, & \text{otherwise} \end{cases}$$

$$Y_{ct} = \begin{cases} 1, & \text{if company } c \text{ is assigned to interval } t \\ 0, & \text{otherwise} \end{cases}$$

P_{dt}	sum of released items across all docks at interval t on day d
Z_{dt}	absolute difference between released and desired items at interval t on day d

4.3.2 Objective Function

The objective of the mathematical model is to ensure that the item flow entering the warehouse is as constant as possible throughout the day by assigning fixed timeslots to a selected number of companies. An item flow is defined as constant if the daily number of items entering the warehouse is equal in every timeslot. We calculate this daily desired number of items by dividing the total daily number of incoming items by the number of time intervals. Subsequently, the absolute difference is calculated between the actual number of incoming items and the desired number of incoming items. The following objective function (4.1) minimizes the total sum of the absolute differences across all days and time intervals:

$$\min \sum_{d=1}^{D} \sum_{t=1}^{T} Z_{dt} \tag{4.1}$$

4.3.3 Constraints

$$\text{s.t.} \quad \sum_{j'=1}^{J} \sum_{t'=t}^{t+proc_j-1} X_{ij'dt'} - 1 \leq M(1 - X_{ijdt}), \quad \forall i \in I, \forall j \in J, \forall d \in D, \forall t \in T, \tag{4.2}$$

$$\sum_{d=1}^{D} \sum_{t=1}^{T} \sum_{i=1}^{I} X_{ijdt} = 1, \quad \forall j \in J, \tag{4.3}$$

$$\sum_{j=1}^{J} X_{ijdt} \leq 1, \quad \forall i \in I, \forall j \in J, \forall t \in T, \tag{4.4}$$

$$X_{ijdt} = 0, \quad \forall i \in I, \forall j \in J, \forall d \in D, \forall t \in T, t \leq |T| - proc_j + 1, \tag{4.5}$$

$$\sum_{t=1}^{T} \sum_{i=1}^{I} X_{ij(DayList_j)t} = 1, \quad \forall j \in J, \tag{4.6}$$

$$\sum_{d=1}^{D} \sum_{i=1}^{I} \sum_{j=1}^{J} X_{ijdt} = Trucks_c Y_{ct}, \quad \forall c \in C, \forall t \in T, TruckToCompany_j = c, \tag{4.7}$$

$$
\begin{aligned}
&Z_{dt} \geq \sum_{j=1}^{J}\sum_{i=1}^{I} X_{ijd(t+proc_j-1)} items_j - df_d, \\
&Z_{dt} \geq df_d - \sum_{j=1}^{J}\sum_{i=1}^{I} X_{ijd(t+proc_j-1)} items_j, \\
&\forall d \in D, \forall t \in T,
\end{aligned}
\tag{4.8}
$$

$$
Z_{dt} \geq 0, \quad \forall d \in D, \forall t \in T, \tag{4.9}
$$

$$
X_{ijdt} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J, \forall d \in D, \forall t \in T, \tag{4.10}
$$

$$
Y_{ct} \in \{0, 1\}, \quad \forall c \in C, \forall t \in T \tag{4.11}
$$

The first constraint (4.2) ensures that there is no overlapping between trucks scheduled on the same dock. The prime indices j and t indicate a different value of the same variable, representing another potential truck to be scheduled. If truck j is assigned to dock i at on day d time t, then it should be impossible to assign another truck to dock i on the same day during the time interval $t + proc_j - 1$. Here, the parameter $proc_j$ represents the total number of intervals it takes to process truck j, accounting for the interval it starts processing. The constraint checks for each combination of dock i, truck j, day d, and time interval t whether it is possible to assign truck j to dock i on day d at time t. The left-hand side of the constraint ensures that at most one truck can be assigned to dock i during the time interval $t + proc_j - 1$, while the right-hand side ensures that the constraint is only enforced if truck j is assigned to dock i on day d at time t.

Constraint (4.3) ensures that each truck is scheduled at least once, while constraint (4.4) limits the number of trucks that can be scheduled per time interval to be at most 1. Constraint (4.5) ensures that no trucks can be scheduled on days during time intervals that exceed their processing time.

Constraint (4.6) fixes that companies only arrive on their designated day with the use of parameter $DayList_j$, which is a list of trucks and their planned day of arrival. The next constraint (4.7 ensures that each company has the same timeslot on every day they arrive. If a certain company c is assigned to a timeslot t and the current truck j is coming from company c, then the binary variable $Y_{ct} =$ will have a value of 1. The right-hand side of the equation will then have the value of the total number of trucks sent by company c defined by the parameter $Trucks_c$. The left-hand side of the equation, which is the sum of trucks assigned to interval t over all docks, trucks, and days, should be equal to this total number of trucks on the right-hand side.

Constraint (4.8) calculates the sum of the number of items released by all trucks across all docks on day d at interval t. This is calculated by adjusting the index t to $t+proc_j-1$ to account for the unloading phase that takes place in the last time interval of the scheduled truck. It also calculates the absolute difference between the number of items released on day d at interval t and the daily desired item flow, serving as a linearization for the absolute objective function (4.1).

Finally, constraint (4.9) enforces the variable Z_{dt} to be greater than or equal to zero, and constraints (4.10) and (4.11) enforce variables X_{ijdt} and Y_{ct} = to be binary with a value of 0 or 1.

4.4 Simulated Annealing

To determine different time slot strategies and assess their effects, we utilize a metaheuristic algorithm called Simulated Annealing. As described in the literature review subsection 3.3.1, the Simulated Annealing algorithm accepts solutions that perform worse than the previous solution during the initial phase. This emphasis on diversification characterizes the early stage of the algorithm, which later transitions into an intensification phase.

The Simulated Annealing algorithm starts by initializing the required data and the cooling scheme settings. It also creates a list of companies that will receive a fixed timeslot that can be adjusted by the algorithm, which is called the *Adjustable-CompanyList*. In addition, a list is created of companies that could potentially be considered for selection to receive a fixed time slot and this is called the *PotentialCompanyList*.

The initial solution of the algorithm is modified with the use of two different operator classes, namely on timeslot level and company level. As long as the temperature remains above a certain threshold, the algorithm iterates, randomly choosing a number between zero and one. Based on this number, a swap, move, or insertion operator is executed on timeslot level. It is also possible for an improved insertion operator to occur. This operator evaluates all the different timeslots for a random company and selects the one that improves the objective value the most. This option becomes available only after a certain number of iterations. This decision is made because the small improved insertion consumes relatively more computational time.

Additionally, after a certain number of iterations, an operator on company level is executed, determining through a random number whether a company is added, removed, or swapped. This operation is not performed in every iteration to explore multiple combinations within a selection of companies. Ultimately, the algorithm checks if the neighbor value is lower than the current value. If this condition is met, the neighbor value becomes the new current value. If not, an acceptance probability and a random number between zero and one are used to determine whether the neighbor value is accepted as the new current value. The probability of acceptance depends on the temperature, with a decreasing temperature resulting in a smaller probability. If the current value is better than the best value found so far, it is stored along with the corresponding schedule. Finally, the temperature is decreased by a decrease factor, and the while loop ends when the temperature falls below the ending temperature.

4.4.1 Initial solution

The development of an initial solution starts with initializing a start schedule. This start schedule is a copy of the historical data that includes all the deliveries that have taken place during the analyzed period, also known as the *scenario data*. Two lists are then created: the *PotentialCompanyList* and the *AdjustableCompanyList*. The minimum and maximum number of companies that can be included in the *AdjustableCompanyList* and those included in the *PotentialCompanyList* are manually determined in advance. Depending on this choice, the minimum number of companies in the *AdjustableCompanyList* is assigned a random timeslot, which is then applied to the start schedule. Subsequently, Algorithm 2 is used for calculating the objective value.

Algorithm 2 begins by creating a daily schedule by extracting all the trucks corresponding to day d from the complete schedule including all days. Then, it calculates the average number of items for that day. Next, for each truck, it determines the timeslot when the unloading process is completed and sums up the associated items within that time interval. Subsequently, it calculates the absolute difference between the number of released items in each time interval and the average number of items(which is the desired flow described in section 4.3). Finally, it sums up all these differences, resulting in the daily objective value. The total objective value is ultimately the sum of all these daily objective values.

Algorithm 2: CalculateObjectiveValue

Function CalculateObjectiveValue(Schedule):

for $d = 1$ **to** D **do**

  $Create\ SubSchedule\ for\ day\ d$;

  $AverageItems\ day\ d \leftarrow \frac{TotalItems\ day\ d}{Intervals}$;

  for $j = 1$ **to** J **do**

    $Calculate\ interval\ when\ truck\ j\ released\ items$;

    $Store\ number\ of\ items\ released\ by\ truck\ j\ in\ this\ interval$

  end

  for $t = 1$ **to** T **do**

    $Difference\ per\ time\ interval\ t \leftarrow |Items\ per\ time\ interval\ t - AverageItems|$

  end

  $DailyObjectiveValue\ of\ day\ d \leftarrow \sum_{t=1}^{T} Difference\ per\ time\ interval\ t$;

end

$TotalObjectiveValue \leftarrow \sum_{d=1}^{D} DailyObjectiveValue\ of\ day\ d$;

Return $TotalObjectiveValue$

4.4.2 Neighbour solution

After calculating an initial value, the simulated annealing algorithm will utilize operators to create neighborhood solutions. These operators perform adjustments at two different levels: the company level and the timeslot level, and the algorithms of these operators are described in Appendix B.

At the company level, three different adjustments are possible:

- Add company: Randomly select a company from the *PotentialCompanyList* and add it to the *AdjustableCompanyList*. This increases the number of companies that receive a company-specific timeslot.
- Remove company: This operator removes a randomly selected company from the *AdjustableCompanyList*. Consequently, the fixed timeslot for this company is eliminated, and the arrival times are reset to the initial state.
- Swap company: Exchange a randomly selected company from the *AdjustableCompanyList* with another randomly selected company from the *PotentialCompanyList*.

Adjustments at the timeslot level can occur in four different ways, exclusively applied to the companies in the *AdjustableCompanyList*:

- Swap timeslots: Swap the designated timeslots of two randomly selected companies.
- Move timeslot: Shift the timeslot of a randomly chosen company up or down by one interval.
- Insert timeslot: Relocate the timeslot of a randomly chosen company to a randomly selected time interval.
- Improved insertion: Almost similar to the insert operator, but this operator evaluates all the different timeslots for a random company and selects the one that improves the objective value the most.

The probabilities for each operator to occur are predetermined and can be adjusted to potentially yield better results.

Both at the company level and the timeslot level, the operators rely on complete randomness, because the selected company is random and the selected timeslot is random. The improved insertion operator does select a random company, but the selected timeslot is based on which insertion improves the current objective value the most.

The reason for implementing this operator is that the SA algorithm has a difficult time finding an improved objective value during the intensification phase, the phase where the temperature decreases. To enhance the discovery of a promising neighborhood, the SA algorithm is equipped with the capability to perform an improved insertion for a timeslot of a randomly selected company after a certain number of iterations.

After generating a neighbor value using the *CalculateObjectiveValue* function, the corresponding schedule is checked for feasibility. The same function keeps track of how many times each time interval is used by all trucks, and if an interval exceeds the limit of 25, it rejects the potential schedule. This limit is based on the maximum number of docks that can be occupied simultaneously.

4.4.3 Cooling scheme

The cooling scheme of the SA algorithm is a schematic overview of the parameter settings that determine how the algorithm behaves. The first parameter is the starting temperature, which governs the level of diversification during the first phase of the algorithm. After each iteration, the temperature is reduced by multiplying it with a decrease factor α, ranging between 0 and 1. A relatively low decrease factor leads to a faster decrease in temperature, reducing the likelihood of accepting worse solutions. The acceptance of a worse solution follows the Boltzmann distribution, which can be expressed as $\frac{-\triangle}{Temp}$ for a minimization problem.

The length of the Markov Chain determines the number of iterations performed at a specific temperature before it is multiplied by the decrease factor. The SA algorithm terminates when a stopping criterion is met, which is reaching an ending temperature in our case.

The pseudocode for our Simulated Annealing Algorithm 3 we have developed is presented on the next page:

Algorithm 3: Simulated Annealing

```
Initialize starting schedule from scenario data;
Initialize cooling scheme settings;
Initialize potential and adjustable companies;
Create initial solution;
while T > T_max do
    for i = 1 to MarkovChainLength do
        if NumberofIterations is modulus of CompanyOperatorValue then
            RandomCompanyNumber ← random.uniform(0, 1);
            if RandomNumber < AddCompanyProb then
                AdjustableCompanyList ← AddCompany;
            else if
              AddCompanyProb < RandomNumber ≤ (AddCompanyProb + RemoveCompanyProb)
              then
                AdjustableCompanyList ← RemoveCompany;
            else
                AdjustableCompanyList ← SwapCompanies;
            end
        end
        RandomNumber ← random.uniform(0, 1);
        if RandomNumber < SwapProbability then
            NeighbourSchedule ← Swap(CurrentSchedule, AdjustableCompanyList);
        else if SwapProbability < RandomNumber ≤ (SwapProbability + MoveProbability) then
            NeighbourSchedule ← Move(CurrentSchedule, AdjustableCompanyList);
        else if ((SwapProbability + MoveProbability) < RandomNumber ≤
          (SwapProbability + MoveProbability + InsertionProbability) then
            NeighbourSchedule ← Insertion(CurrentSchedule, AdjustableCompanyList);
        else
            if NumberofIterations > LocalSearchIterations then
                NeighbourSchedule ← LocalSearch(CurrentSchedule, AdjustableCompanyList)
            end

        end
        NeighbourValue ← CalculateObjectiveValue(TemporarySchedule);
        Check feasibility;
        Delta ← NeighbourValue − BestValue;
        if AcceptanceProbability > random.uniform(0, 1) then
            CurrentSchedule ← NeighbourSchedule;
            CurrentValue ← NeighbourValue;
        end
        if CurrentValue < BestValue then
            BestSchedule ← CurrentSchedule;
            BestValue ← CurrentValue;
        end
    end
    T ← αT;
end
```

4.5 Conclusion

We present a Simulated Annealing heuristic which is able to assign fixed timeslots to a selected group of companies in order to minimize the absolute differences between the desired number of incoming items and the actual number of incoming items. Our truck scheduling problem is defined as a Mixed Integer Linear Program model and that forms the foundation of our Simulated Annealing algorithm.

We create an initial solution schedule by randomly assigning a different fixed timeslot to a number of companies based on our algorithm settings. By executing different operators on a timeslot level and company level, we generate neighbour solutions that will be accepted based on the current solution value and the temperature. The feasibility of the solution is checked by verifying if the number of trucks in a time interval exceeds the maximum dock limit.

Chapter 5

Experiment Design

This chapter addresses the subquestion "How should the experimental phase be designed?". The first section 5.1 outlines the technical specifications for conducting the experiments. Section 5.2 explains how we utilize the company data as input for our model, while the third section 5.3 explains the entire process of calibrating the SA parameters and presents the resulting cooling scheme. Subsequently, section 5.4 elaborates on the experiments conducted with the tuned algorithm. Finally, the key aspects of this chapter are summarized in section 5.5.

5.1 Technical specifications

The experiments were conducted on a computer equipped with 16GB RAM and an Intel Core i7 processor with a speed of 2.8 GHz. The Simulated Annealing algorithm was implemented using Python 3.9 language within the Spyder 5.4.2 IDE.

5.2 Input data

For the model calculations, historical dataset dating from March 3rd, 2022 to October 28th, 2022 is used. This data was chosen because, at the time of analysis, it was the most complete data available. Data prior to this time frame was not complete due to the retention policy, as also mentioned in subsection 2.2.1. For each delivery, the company name, date of arrival, time of arrival, number of pallets delivered, and number of items delivered were retrieved from the database.

In order to utilize the acquired data from the database in both the MILP model and the simulated annealing algorithm, certain data need to be reformulated. The date of arrival is transformed into a day number, where 3-3-2022 represents the first day and 28-10-2022 represents the last day, denoted as day 168. All deliveries within the same day will be categorized into one of the 11 daily time intervals based on their arrival time. For instance, a delivered shipment at 9:30 a.m. becomes time interval 3.

The completion time of a job depends on the number of pallets carried by that truck. A rule of thumb for estimating the total processing time is that the constant time per truck is 10 minutes, and each additional pallet adds 1 minute to it.

5.3 Algorithm tuning

To tune the parameters of the algorithm, we adopted a fixed scenario, allowing for the comparison of outcomes across different runs. For this purpose, we selected 20 companies to be assigned fixed time slots, which could also be adjusted. These companies were chosen based on the total number of items delivered annually. Collectively, these 20 companies account for 48.3% of the total items delivered, thus ensuring a significant impact on the objective value when modifying the time slots. Before initiating the tuning process, we decided to assess the computational times associated with all operators and the calculation of the objective value. These times are presented in Table 5.1 and can be taken into consideration during the selection process of the algorithm parameters. Three of the timeslot operators, swap, move, and insertion, also include the computational time of the objective value calculation which occurs after the operation. We observe that the time required to evaluate a solution is 0.127 seconds, and the time it takes to compute a neighbouring solution is insignificant.

Table 5.1: SA operator computation times

Level	Operator	Time (sec)
Timeslot	Swap	0.129
	Move	0.128
	Insertion	0.128
	Impr. insert.	1.565
Company	Add	0.001
	Remove	0.001
	Swap	0.002
Model	Objective Value	0.127

5.3.1 Cooling scheme settings

Starting - and ending temperature

The first two parameters of the Simulated Annealing cooling scheme are the start temperature and the ending temperature. By setting a very high start temperature and plotting the temperature against the acceptance ratio, Figures 5.1 and 5.2 are generated. The acceptance ratio is calculated by dividing the negative delta of the temporary - and the best solution with the current temperature and then taking the exponent of this division.

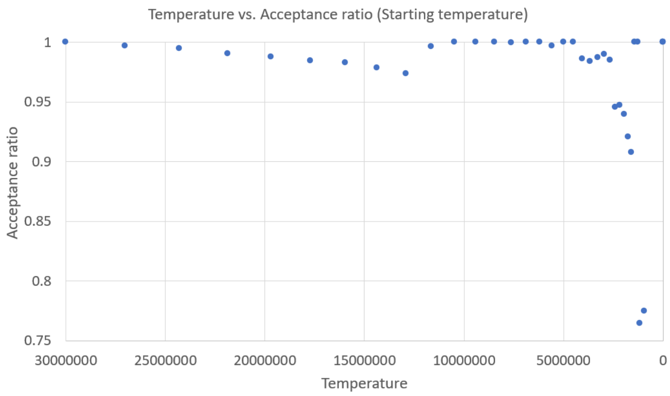

Figure 5.1: Starting temperature calibration

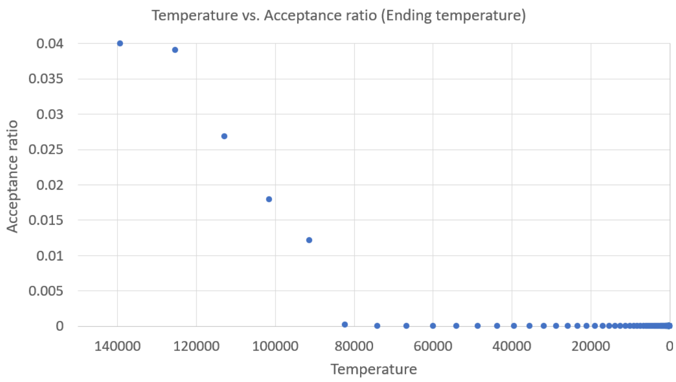

Figure 5.2: Ending temperature calibration

The left Figure 5.1 illustrates that starting from a temperature of approximately 2,800,000, the acceptance ratio consistently decreases and falls below a value of 1. The right Figure 5.2 displays the final phase of the SA algorithm, revealing that the acceptance ratio approaches nearly 0 around a value of 82,000. Based on these observations, the chosen starting temperature is 2,800,000 and the ending temperature is 82,000. These values will also be applied to the remaining parameter calibrations.

Decrease factor and length Markov Chain

In order to determine the decrease factor and the length of the Markov Chain, we conducted experiments using different settings and evaluated the computational time and objective value for each setting. To mitigate the randomness of the outcomes, each setting was executed five times, and the average was taken. The results of the various decrease factor settings are presented in the following Table 5.2.

Table 5.2: Alpha factor calibration

		Setting 1	Setting 2	Setting 3	Setting 4
Alpha tuning	**Alpha**	0.6	0.7	0.8	0.9
	Markov Chain	10	10	10	10
Computational Time (sec)	Run 1	*9.49*	*13.29*	*22.04*	*45.44*
	Run 2	*9.52*	*13.5*	*21.92*	*45.47*
	Run 3	*9.9*	*13.55*	*21.62*	*47.62*
	Run 4	*9.36*	*13.27*	*21.87*	*45.43*
	Run 5	*9.38*	*13.83*	*21.54*	*44.91*
	Average Time	**9.63667**	**13.4467**	**21.86**	**46.1767**
Objective Value	Run 1	*7448879*	*7319323*	*7345155*	*7251373*
	Run 2	*7543483*	*7246025*	*7268771*	*7091549*
	Run 3	*7381683*	*7454571*	*7589535*	*7186829*
	Run 4	*7586099*	*7418467*	*7344543*	*7274237*
	Run 5	*7189806*	*7204468*	*7184445*	*7167963*
	Average Value	**7429990**	**7328571**	**7346490**	**7194390**

An alpha value of 0.9 yields the best objective values, but it is associated with a significantly higher average computational time. Considering the potential additional time required for adjusting the length of the Markov Chain, we decided not to select settings 3 and 4. Since the objective values for the remaining alpha values are relatively close, and the average computational time between 9 and 14 seconds is also acceptable, we chose alpha values of 0.6 and 0.7 to proceed with further experimentation. This involves increasing the Markov Chain length in combination with the two chosen alpha settings, and the results of this tuning are shown in Table 5.3:

Table 5.3: Length Markov Chain calibration

		Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6	Setting 7	Setting 8	Setting 9	Setting 10
Markov Chain tuning	**Alpha**	0.6	0.6	0.6	0.6	0.6	0.6	0.7	0.7	0.7	0.7
	Markov Chain	20	30	40	50	60	70	20	30	40	50
Computational Time (sec)	Run 1	*18.69*	*27.82*	*37.01*	*46.51*	*60.51*	*69.29*	*27.14*	*41.44*	*57.75*	*75.94*
	Run 2	*18.72*	*28.04*	*38.37*	*48.03*	*55.56*	*69.31*	*26.83*	*42.41*	*57.6*	*76.04*
	Run 3	*18.85*	*28.46*	*38.75*	*47.23*	*58.26*	*64.82*	*28.13*	*42.15*	*65.69*	*74.89*
	Run 4	*18.72*	*28.94*	*36.68*	*46.78*	*60.97*	*67.64*	*26.79*	*44.9*	*73.93*	*65.88*
	Run 5	*18.99*	*29.61*	*37.38*	*46.93*	*58.07*	*70.09*	*27.11*	*39.48*	*54.62*	*70.5*
	Average Time	**18.794**	**28.574**	**37.638**	**47.096**	**58.674**	**68.23**	**27.2**	**42.076**	**61.918**	**72.65**
Objective Value	Run 1	*7143991*	*7352541*	*7237971*	*7183169*	*7278827*	*7176371*	*7206651*	*7162879*	*7195583*	*7162401*
	Run 2	*7369553*	*7192029*	*7200227*	*7283577*	*7136429*	*7177925*	*7238085*	*7240501*	*7244789*	*7196263*
	Run 3	*7279925*	*7383433*	*7262879*	*7141981*	*7062665*	*7204799*	*7260423*	*7273987*	*7025739*	*7224743*
	Run 4	*7352013*	*7355921*	*7220809*	*7287755*	*7278663*	*7293421*	*7133259*	*7203523*	*7274717*	*7226663*
	Run 5	*7310133*	*7206659*	*7105207*	*7198753*	*7138811*	*7258294*	*7460281*	*7134401*	*7302185*	*7145287*
	Average Value	**7291123**	**7298117**	**7205419**	**7219047**	**7179079**	**7222162**	**7259740**	**7203058**	**7208603**	**7191071**

It is observed that increasing the length of the Markov Chain leads to more stable objective values. Nevertheless, a longer Markov Chain results in an increase in computational time. It is important to consider that we still need to calibrate the operator settings, including the improved insertion operator. Compared to the other operators (Table 5.1), the improved insertion operator requires more time to execute, emphasizing the need for headroom in the computational time when choosing the Markov Chain length. Ultimately, we selected an alpha value of 0.6 and a Markov Chain length of 40. This setting yields relatively low and stable objective values, allows sufficient computational time for the improved operator, and ensures an adequate number of iterations for exploring new combinations of companies, for instance.

5.3.2 Operator settings

The Simulated Annealing algorithm operators can be divided into two levels: the timeslot level and the company level. In order to maintain a consistent number of adjustable companies for more reliable experimental outcomes, we decided to start by calibrating the timeslot operators.

Timeslot operators

Within this category, we chose to focus on the three divergent operators: swap, move, and insertion. Based on the resulting average computational time, we can make a more informed decision regarding when the algorithm can incorporate the improved insertion operator, as this incurs relatively high computational time per iteration.

Table 5.4: Timeslot operators calibration

Timeslot operators tuning		Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6	Setting 7	Setting 8	Setting 9	Setting 10
	Swap	1	0	0	0.5	0	0.5	0.333	0.4	0.4	0.2
	Move	0	1	0	0.5	0.5	0	0.333	0.4	0.2	0.4
	Insertion	0	0	1	0	0.5	0.5	0.333	0.2	0.4	0.4
Computational Time (sec)	Run 1	*37.21*	*37.7*	*36.82*	*36.82*	*36.37*	*36.83*	*37.13*	*36.64*	*38.71*	*37.41*
	Run 2	*37.56*	*37.4*	*40.08*	*37.41*	*37.16*	*36.92*	*37.32*	*37.44*	*40.34*	*38.83*
	Run 3	*37.86*	*42.96*	*37.95*	*39.36*	*38.55*	*37.39*	*38.21*	*40.74*	*41.71*	*39.8*
	Run 4	*39.61*	*41.38*	*40.23*	*39.19*	*39.43*	*38.9*	*39.74*	*37.12*	*40.03*	*39.3*
	Run 5	*39.67*	*39.73*	*39.2*	*36.67*	*39.13*	*40.76*	*39.28*	*38.95*	*36.84*	*38.87*
	Average Time	**38.382**	**39.834**	**38.856**	**37.89**	**38.128**	**38.16**	**38.336**	**38.178**	**39.526**	**38.842**
Objective Value	Run 1	*7648061*	*7614761*	*7252103*	*7430269*	*7209821*	*7305953*	*7190795*	*7311753*	*7221275*	*7114881*
	Run 2	*7636073*	*7337617*	*7213029*	*7282307*	*7248903*	*7203455*	*7099289*	*7292643*	*7230611*	*7294679*
	Run 3	*7544013*	*7256251*	*7269717*	*7253843*	*7260775*	*7250133*	*7354099*	*7178937*	*7203431*	*7302929*
	Run 4	*7753473*	*7581203*	*7221959*	*7493319*	*7110115*	*7244151*	*7112685*	*7281149*	*7328129*	*7197509*
	Run 5	*7609991*	*7250457*	*7160845*	*7495589*	*7296997*	*7190589*	*7254759*	*7370575*	*7159927*	*7299757*
	Average Value	**7638322**	**7408058**	**7223531**	**7391065**	**7225322.2**	**7238856**	**7202325**	**7287011**	**7228675**	**7241951**

The results of the experiments with different settings are shown in the above Table 5.4. Firstly, it is evident that settings 1 and 2 perform the poorest while setting 3, which solely utilizes the insertion operator, performs relatively well. However, it should be noted that excluding operators may result in a lower divergence rate, potentially missing out on good neighbourhood values. Therefore, we prefer a setting where all three operators have a chance to be selected. The average computational time per setting is nearly equal across all settings, which aligns with our expectations based on Table 5.1. Ultimately, we selected setting 7 due to its relatively low and stable objective values.

Subsequently, we incorporated the improved insertion operator probability and conducted experiments to assess the algorithm's performance under different settings. It is crucial to strike a balance between computational time and objective value in order to achieve optimal results. The outcomes of these calibration experiments can be found in the following Table 5.5:

Table 5.5: Improved insertion calibration

		Setting 1	Setting 2	Setting 3	Setting 4
Local search operator tuning	**Swap**	0.333	0.3	0.2833	0.2667
	Move	0.333	0.3	0.2833	0.2667
	Insertion	0.333	0.3	0.2833	0.2667
	Local search	0	0.1	0.15	0.2
Computational Time (sec)	Run 1	*37.13*	*90.56*	*89.26*	*115.76*
	Run 2	*37.32*	*86.51*	*105.77*	*144.03*
	Run 3	*38.21*	*85.09*	*116.78*	*139.25*
	Run 4	*39.74*	*74.78*	*96.32*	*149.15*
	Run 5	*39.28*	*83.72*	*121.38*	*128.93*
	Average Time	**38.336**	**84.132**	**105.902**	**135.424**
Objective Value	Run 1	*7190795*	*7065435*	*7127531*	*7037587*
	Run 2	*7099289*	*7052591*	*7024175*	*7008501*
	Run 3	*7354099*	*7107171*	*7133591*	*7172439*
	Run 4	*7112685*	*7147551*	*7224615*	*7069005*
	Run 5	*7254759*	*7211591*	*7150901*	*7019805*
	Average Value	**7202325**	**7116868**	**7132163**	**7061467**

It is immediately noticeable that the objective values improve compared to the results from Table 5.4. However, the computational time significantly increases as the probability of executing the improved insertion operator increases. As we aim for good results while keeping the computational time within acceptable limits, we choose to apply setting 4 but not from the start of the algorithm. Under these settings, the SA algorithm performs 280 iterations until it reaches the ending temperature of 82.000, and we conducted the following experiments to determine the optimal number of iterations at which we want to activate the improved insertion operator:

Table 5.6: Improved insertion iterations

		Setting 1	Setting 2	Setting 3	Setting 4
Local search iterations	**Number of iterations**	100	150	200	250
Computational Time (sec)	Run 1	*99.67*	*68.35*	*67.4*	*43.18*
	Run 2	*104.19*	*68.61*	*63.89*	*56.08*
	Run 3	*93.47*	*86.04*	*66.08*	*48.07*
	Run 4	*116.2*	*78.73*	*70.87*	*50.81*
	Run 5	*105.86*	*85.21*	*71.72*	*49.21*
	Average Time	**103.878**	**77.388**	**67.992**	**49.47**
Objective Value	Run 1	*7076681*	*7084341*	*7092897*	*7007503*
	Run 2	*7128107*	*7126461*	*7136495*	*7102725*
	Run 3	*7061303*	*7050865*	*7048567*	*7397641*
	Run 4	*7029791*	*7174455*	*7073203*	*7027605*
	Run 5	*7153303*	*7040067*	*7135445*	*7212841*
	Average Value	**7089837**	**7095238**	**7097321**	**7149663**

Table 5.6 shows that the average value for each setting is approximately the same, except for the last setting 4. However, the average time per setting decreases as the improved insertion is less frequently executed, which saves relatively more time. Ultimately, we selected setting 2, where the improved insertion operator is added after 150 iterations. From this number of iterations onwards, the algorithm transitions towards intensification, and we want to stimulate this process by enabling the option of improved insertion.

Company operators

Unlike the timeslot operators, the company operators depend on the number of iterations performed. For the company operators, we have chosen to apply them after a fixed number of iterations in order to explore different combinations of companies. The reason for this is that after changing the possible companies that receive a timeslot, we want to allow a sufficient number of iterations with that selection to take place. We determined this fixed number of iterations through several experiments, where a minimum of 20 companies could be adjusted and a maximum of 30:

Table 5.7: Company operators iterations

Company operators		**Setting 1**	**Setting 2**	**Setting 3**
	Number of iterations	5	10	15
Computational Time (sec)	Run 1	*78.35*	*73.86*	*88.77*
	Run 2	*78.1*	*80.69*	*89*
	Run 3	*78.88*	*95.9*	*92.15*
	Run 4	*71*	*93.3*	*69.27*
	Run 5	*85.51*	*82.84*	*95.08*
	Average Time	**78.368**	**85.318**	**86.854**
Objective Value	Run 1	*6985251*	*7019885*	*7267437*
	Run 2	*7164435*	*7331315*	*7172967*
	Run 3	*7224821*	*7215775*	*7078457*
	Run 4	*7260035*	*7126527*	*7245095*
	Run 5	*6996233*	*7062201*	*7176957*
	Average Value	**7126155**	**7151141**	**7188183**

Table 5.7 shows three different settings where the number of iterations stands for each moment a company operator is selected. So for example a number of iterations with a value of five means that after each five iterations, a company operator is selected. From Table 5.7, it is evident that the first setting performs the best in terms of the lowest average computation time. This is why we have chosen this setting, which means that the algorithm will execute a company operator every 5 iterations. The probabilities associated with each of the three different operators being executed are determined in the following Table 5.8.

Table 5.8: Company operators probabilities

		Setting 1	Setting 2	Setting 3	Setting 4
Company operators probabilities	**Add company**	0.333	0.4	0.4	0.2
	Remove company	0.333	0.4	0.2	0.4
	Swap Companies	0.333	0.2	0.4	0.4
Computational Time (sec)	Run 1	*76.24*	*83.1*	*79.79*	*85.66*
	Run 2	*81.04*	*83.27*	*85.28*	*92.24*
	Run 3	*86.41*	*76.7*	*94.23*	*82.48*
	Run 4	*107.17*	*91.93*	*80.3*	*88.8*
	Run 5	*94.57*	*79.98*	*83.73*	*73.19*
	Average Time	**89.086**	**82.996**	**84.666**	**84.474**
Objective Value	Run 1	*6906007*	*7206021*	*7160207*	*7132323*
	Run 2	*7177933*	*7091347*	*7102373*	*7190897*
	Run 3	*7075293*	*7189289*	*7128263*	*7116565*
	Run 4	*7152623*	*7113111*	*7094783*	*7092467*
	Run 5	*7217685*	*7234917*	*7043099*	*7208929*
	Average Value	**7105908**	**7166937**	**7105745**	**7148236**

Table 5.8 indicates that all settings achieve similar scores in terms of average objective value. We have ultimately selected setting 1, as it yielded the highest individual objective value among these settings.

Cooling scheme

The complete cooling scheme, determined after conducting all the experiments, is shown in Table 5.9:

Table 5.9: Cooling scheme

Parameter	Setting	After 150 iterations
Start temperature	2800000	2800000
End temperature	82000	82000
Decrease factor	0.6	0.6
Markov Chain length	40	40
Swap	0.3333	0.2667
Move	0.3333	0.2667
Insertion	0.3333	0.2667
Impr. insert.	0	0.2
Company iterations	5	5
Add company	0.333	0.333
Remove company	0.333	0.333
Swap companies	0.333	0.333

5.4 Experimental phase

After calibrating all parameters of the Simulated Annealing algorithm, we intend to utilize the algorithm to conduct various experiments, which will be briefly described in this section.

- **Baseline analysis**: The first experiment aims to compare the performance of our Simulated Annealing algorithm by analyzing the results of the baseline algorithm in relation to current objectives, as well as the development of the WIP amount. This analysis helps us understand how our algorithm performs.
- **Model validation**: In the next experiment, we compare our algorithm with three predefined strategies: a random strategy where companies are assigned random timeslots, an item-based strategy where companies are sorted based on the number of items they have delivered, and a frequency-based strategy where companies are sorted based on their arrival frequency. In the latter two strategies, the first company receives timeslot number one, the second company receives timeslot number two, and so on until timeslot eleven is reached. After that, the timeslot allocation starts again from timeslot number one. This experiment aims to determine if our algorithm outperforms less intelligent predefined strategies.
- **Sensitivity analysis**: The third experiment tests the impact of the number of companies added to the solution to examine its influence on the objective value and the development of the WIP. We conduct this experiment to determine the point at which adding more companies is no longer profitable.
- **Timeslot strategy scoring**: Finally, we score different timeslot strategies derived from varying numbers of companies and compare them with each other, as well as with the current solutions.

5.5 Conclusion

This chapter focused on describing the technical specifications of the Simulated Annealing algorithm to conduct our experiments. Additionally, it provided an overview of how we utilize historical data as input for our model. Furthermore, this chapter also provided a description of the tuning of the parameters of the Simulated Annealing algorithm and elaborated on the experiments conducted with this cooling scheme. For this purpose, multiple runs were conducted to ensure the selection of the appropriate settings, resulting in a clear understanding of the performance of all operators. The Simulated Annealing algorithm starts with a diversification phase where random company and timeslot operators are executed to generate neighbour solutions. As the temperature decreases, the algorithm automatically diverges more and the probability of selecting a worse solution decreases. The process of diverging is aided by an improved insertion operator which becomes available for the algorithm to use after a certain number of iterations. Finally, we provided a description of all the experiments, with their respective results presented in the following chapter 6.

Chapter 6

Experimental Results

In this chapter, the results of the experiments described in the previous chapter 5 are presented and the question "How does our algorithm perform?" is answered. The first section 6.1 provides a detailed account of the baseline analysis we conducted, which involved comparing the baseline solution with the optimal and current objective values, as well as examining the development of the WIP. The second section 6.2 delves into the validation of our algorithm, while the third section 6.3 elaborates on the relationship between the number of companies in the solution and the objective value, as well as the WIP development. Finally, section 6.4 presents the scores obtained from different timeslot strategies, and section 6.5 provides a conclusion.

6.1 Algorithm analysis

Within this section, we compare the performance of our Simulated Annealing algorithm solution with the current solution performance. During the execution of the experiments, we discovered errors in the data. Specifically, there were shipments with an unusually large number of items, which resulted in a skewed distribution of the WIP values. After cross-referencing these shipments with the company's database, we found that the item quantities had been incorrectly entered. We corrected these quantities to reflect the accurate number of items. As a result, the objective values in the upcoming sections are lower than those reported in chapter 5.

6.1.1 Baseline results

We have expressed the results of executing the tuned Simulated Annealing algorithm in two different ways: the objective value as described in subsection 4.3.2, and the average Work-in-Progress per day. The average WIP per day was calculated by summing the deviations, both negative and positive, within the same time slot across all days and dividing it by the total number of days. This provides an average approximation of the number of items that are over or under-received compared with the average service rate for each time slot. The average daily service rate is dependent on the total number of items expected to be received. This value is then divided by the number of hours a day, and the employees are scheduled accordingly. The results of this analysis can be observed in Figures 6.1 and 6.2.

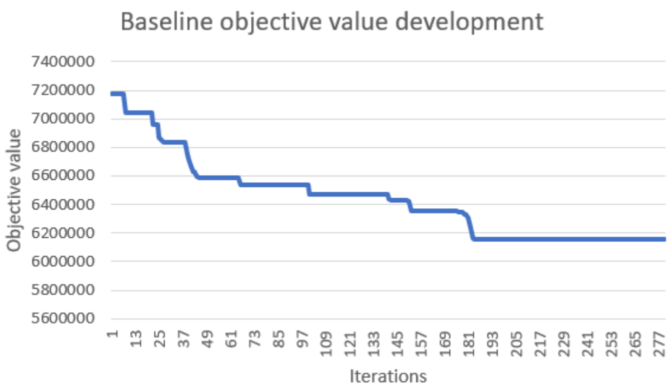

Figure 6.1: Baseline objective value development

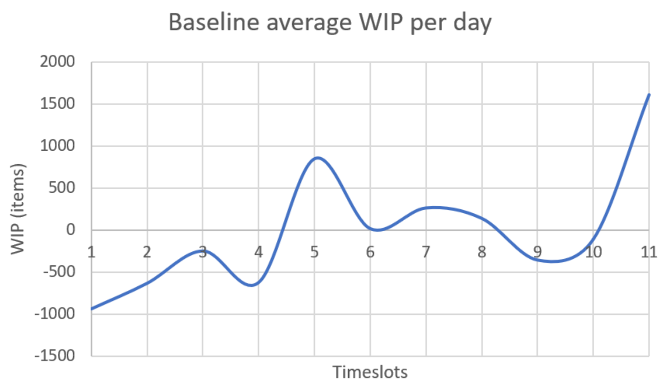

Figure 6.2: Baseline average WIP per day

Figure 6.1 demonstrates that as the iterations increase, the objective value decreases until it reaches a value of 6,156,372 items after 184 iterations. The right figure 6.2, illustrates the variation in the average WIP per day for this baseline solution. It is noteworthy that the WIP values in the first and last time intervals are relatively low and high compared to the other time slots. One explanation for this is that the algorithm struggles to fill the first time slot with items, as shipments with a relatively large number of items also take longer to process. Consequently, the completion time for these shipments often falls outside the first time slot, resulting in a negative average WIP. The elevated value in the last time slot can be attributed to shipments that span across the last time slot. Due to the fixed time slot allocated to companies with relatively large shipments, there are days when the processing time of a shipment exceeds the last time slot. In such cases, the items are attributed to the last time slot. This limitation of the model should be taken into account when analyzing the WIP. Both Figure 6.1 and Figure 6.2 are challenging to analyze without context, thus in the following subsection, we will provide context by comparing them to the current situation.

Table 6.1 indicates that the model has selected a total of 27 companies to allocate time slots, within a maximum limit of 30 companies and a minimum of 20.

Table 6.1: Baseline solution companies

1	2	3	4	5	6	7	8	9	10	11
H	WL	ZF		CL	VH	UK	Y	JI	ACA	UP
HV		CV			AA	BE	VJ	PG	M	CM
DE		XE			CH			ACN	US	PA
UJ		OP							PM	
									X	

6.1.2 Comparison with current situation

To obtain a clear understanding of the performance of the Simulated Annealing algorithm, we compared the objective value with the performance of the current situation. The current situation was derived by making no adjustments to the historical data.

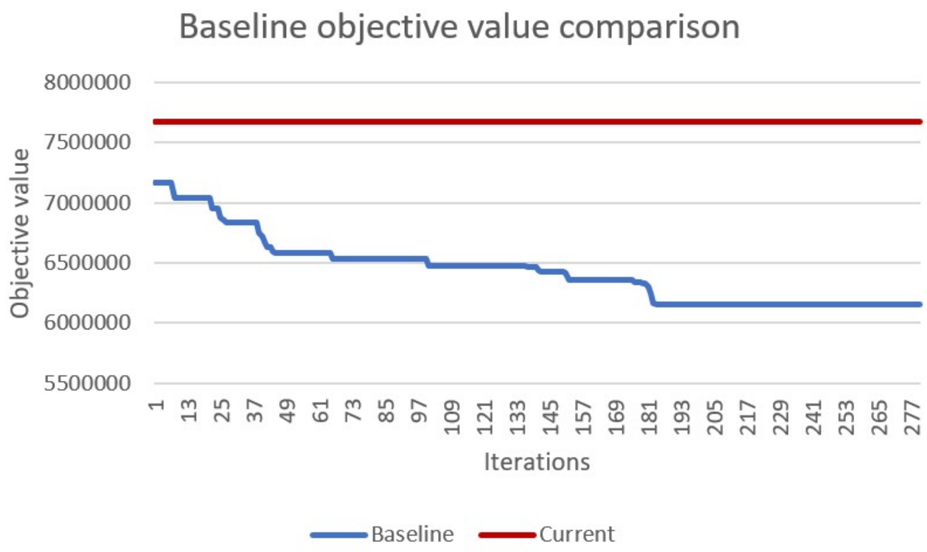

Figure 6.3: Baseline objective value comparison

	Current	Baseline	Difference	
Objective value	7673278	6156372	-1516906	-19.77%
Average value	45674	36645	-9029	

Table 6.2: Baseline comparison with optimal and current situation

Figure 6.3 illustrates the development of the baseline objective value, gradually moving away from the current objective value. It is important to note that the baseline value does not start at the same point as the current situation value, as the initial value of the Simulated Annealing algorithm is a selection of 20 companies, each assigned a random time slot, resulting in an initial improvement.

Table 6.2 presents an analysis of the objective value improvement, which has also been converted into an average value by dividing the objective value by the total number of days. The difference in items between the current situation and the baseline situation is 1,516,906, which is a percentage improvement of -19.77%.

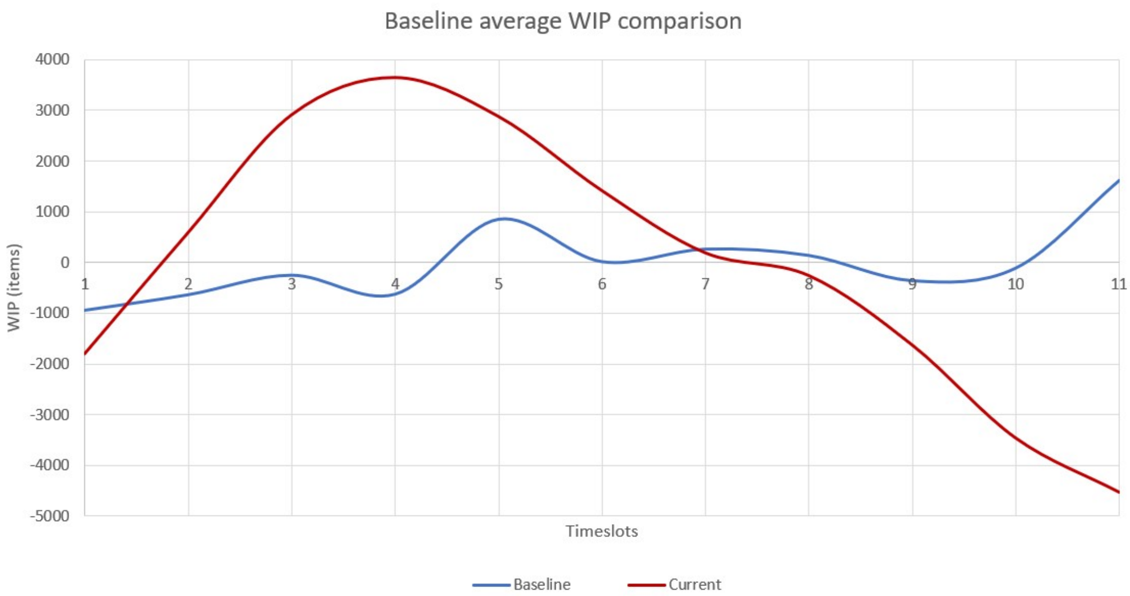

Figure 6.4: Baseline average WIP comparison

Comparing the average WIP values of the SA baseline and the current situation results in the above Figure 6.4, which clearly demonstrates the improvement in WIP. The WIP pattern in the current situation starts with a negative value and reaches a peak of +3,649 items in the fourth time slot as the morning hours progress. It then steadily declines until it reaches a minimum of -4,523 items in the last time slot. The baseline solution shows a significantly improved WIP pattern compared to the current situation, exhibiting much more consistency. To analyze the differences between the current solution and the baseline solution, we created Figure 6.5 and Table 6.3:

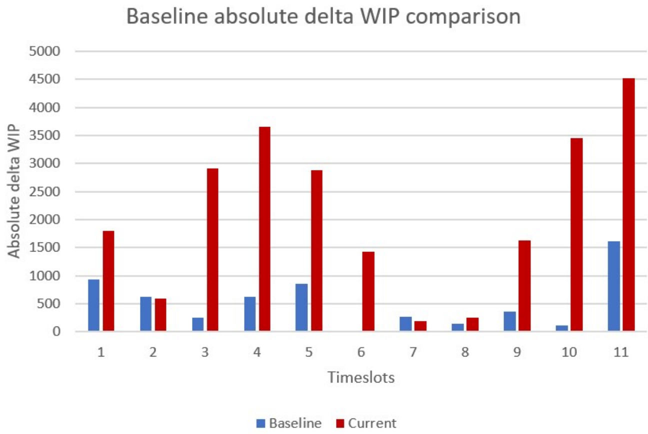

Figure 6.5: Baseline delta WIP comparison

Timeslot	Current	Baseline	Difference	
1	1794	932	-862	-48.05%
2	593	628	+35	+5.87%
3	2914	245	-2669	-91.59%
4	3649	618	-3031	-83.07%
5	2882	850	-2032	-70.51%
6	1421	20	-1401	-98.59%
7	194	266	+72	+36.99%
8	254	142	-112	-44.04%
9	1623	353	-1270	-78.25%
10	3456	106	-3350	-96.93%
11	4523	1610	-2913	-64.40%

Table 6.3: Baseline absolute WIP comparison

Figure 6.5 confirms our conclusion that the average WIP value of the baseline solution has significantly improved compared to the current situation. In Table 6.3, it can be observed that in most cases, the WIP has decreased by percentages ranging from -44.04% to -98.59%. In the two cases where the WIP has increased, the absolute difference is 35 items and 72 items, which is relatively small.

6.2 Model validation

To validate the performance of the Simulated Annealing algorithm utilized in our model, we will compare the outcomes of the objective value and the average WIP with three predefined strategies, namely:

- Random strategy: In this strategy, all companies are assigned a random time slot.

- Item-based strategy: In this strategy, all companies are sorted from largest to smallest based on the number of items they have delivered over all days. Subsequently, the first company is assigned time slot 1, the second company is assigned time slot 2, and so on until time slot 11 is reached, after which the counting restarts.

- Frequency-based strategy: In this strategy, all companies are sorted from largest to smallest based on the number of arrivals over all days. Subsequently, the first company is assigned time slot 1, the second company is assigned time slot 2, and so on until time slot 11 is reached, after which the counting restarts.

All of these strategies will be applied to the same set of companies as those selected in the baseline solution (Table 6.1). The comparison of objective values can be observed in the following Figure 6.6 and Figure 6.7:

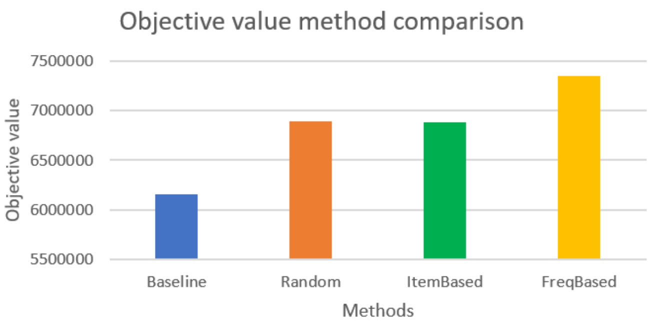

Figure 6.6: Objective value strategies comparison

	Baseline	Random	ItemBased	FreqBased
Objective value	6156372	6894848	6883950	7356338
Average value	36645	41041	40976	43788
Difference		+4396 +12%	+4331 +11.82%	+7143 +19.49%

Figure 6.7: Objective values percentual differences

Based on the objective value, it is evident that the baseline solution has the lowest value compared to the other three predefined strategies. It performs better than the random strategy (+12%), the item-based strategy (+11.82%), and the frequency-based strategy (+19.49%). To gain further insight into the variation of the average Work-in-Progress, the following graph have been generated:

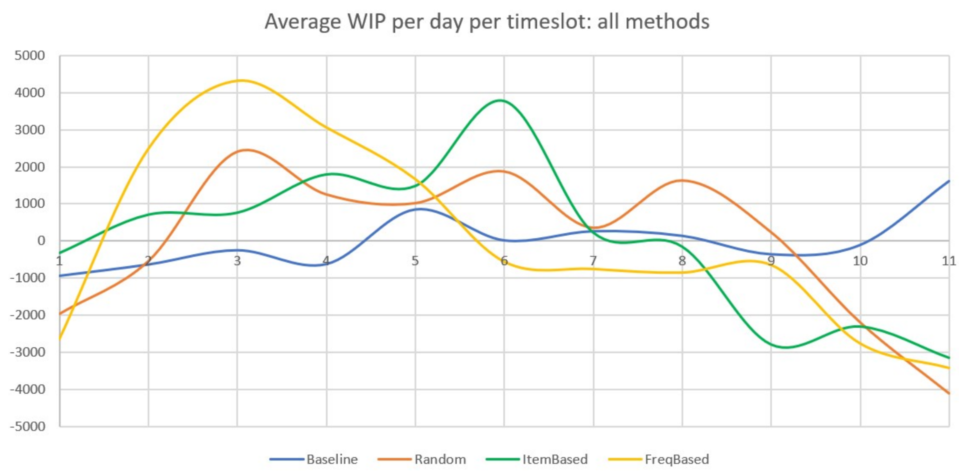

Figure 6.8: Baseline vs. strategies WIP comparison

In Figure 6.8 it can be observed that there is a relatively low dip in terms of WIP items during the morning hours. This can be attributed to the fact that the WIP in the current situation also exhibits a low dip in the morning. The random strategy slightly mitigates this dip, but not completely, while the frequency-based companies, due to their relatively lower number of delivered items, have less impact, resulting in a marginal decrease of the peak. Figure 6.8 illustrates that the WIP increases during the morning hours until a peak is reached in the middle of the day. Subsequently, the WIP decreases, reaching its lowest point in the final time slot. This low value in the last time slot corresponds to the pattern of WIP in the current situation, as observed in Figure 6.4.

These graphs demonstrate that the intelligent strategy of the Simulated Annealing algorithm outperforms the predefined strategies.

6.3 Sensitivity analysis

To investigate the impact of the number of companies assigned a fixed time slot on the objective value, we conducted experiments where we incrementally added 5 companies before we ran the Simulated Annealing algorithm. In these experiments, we doubled the length of the Markov Chain to enhance the probability of obtaining a good solution with the increased number of companies.

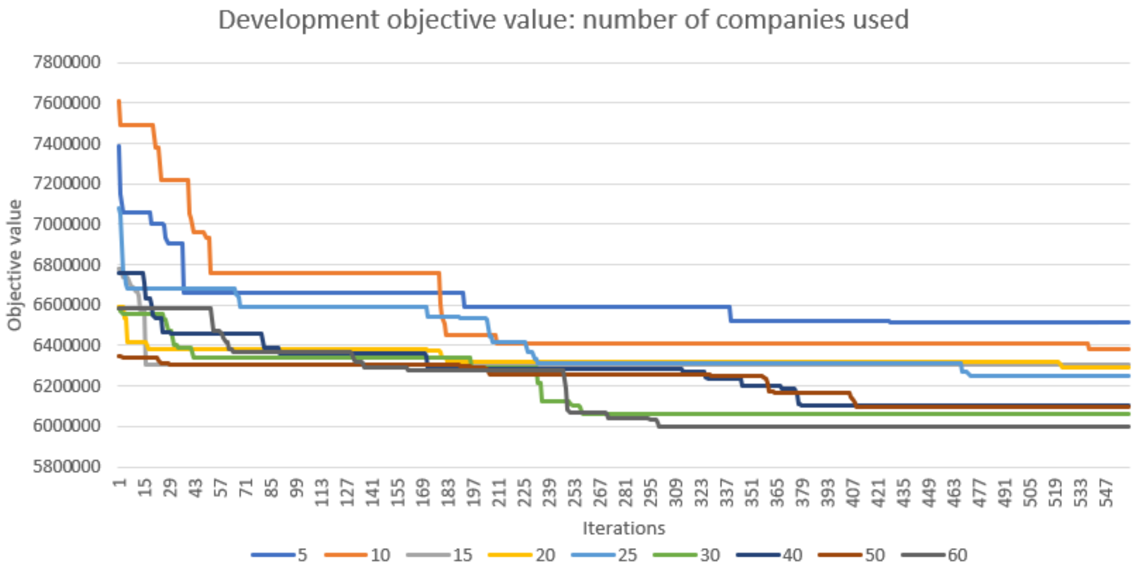

Figure 6.9: Development objective value: number of companies

It is evident from Figure 6.9 that as the number of companies assigned a fixed time slot increases, the objective value decreases. When 5 companies have a fixed time slot, the objective value is 6,516,056 (with an average of 38,786), whereas, with 60 companies, the value is 5,999,212 (with an average of 35,710). It becomes clear that adding additional companies reaches a point where the relatively small improvement in objective value does not outweigh the added complexity.

Additionally, we examine the effect of adding more companies to the solution on the development of the daily average WIP, and these results are presented in appendix C. The key observation from these results is that despite the decrease in objective value, the development of the WIP during the day does not become more constant. Although there is a clear enhancement in the WIP pattern when comparing the results of the current and baseline situations, the objective of the algorithm is to decrease the total number of items in the WIP, so it is no guarantee that the WIP becomes a straight line. This observation is supported by Table 6.4:

Table 6.4: Calculation objective value and WIP

		Timeslots					Daily objective value
		1	2	...	m-1	m	
Days	1	(1,1)	(2,1)	...	(m-1,1)	(m,1)	$\Sigma_{m=1}$(m,1)
	2	(1,2)	(2,2)	...	(m-1,2)	(m,2)	$\Sigma_{m=1}$(m,2)
	⋮	⋮	⋮		⋮	⋮	⋮
	n-1	(1,n-1)	(2,n-1)	...	(m-1,n-1)	(m,n-1)	$\Sigma_{m=1}$(m,n-1)
	n	(1,n)	(2,n)	...	(m-1,n)	(m,n)	$\Sigma_{m=1}$(m,n)
WIP per timeslot		$\Sigma_{n=1}$(1,n)	$\Sigma_{n=1}$(2,n)	...	$\Sigma_{n=1}$(m-1,n)	$\Sigma_{n=1}$(m,n)	

Table 6.4 presents a schematic representation of the calculation of the daily ob-

jective value and the WIP per timeslot. The significant difference lies in the fact that the objective value is obtained by summing the values across all timeslots for each day and then aggregating these daily values. On the other hand, the WIP is computed by summing the values per timeslot across all days. In the model, the objective value serves as the guiding metric, prompting the algorithm to minimize the total deviation on a daily basis. This does not imply that all timeslots will exhibit an equal deviation, as there can be variations and distribution among them. For instance, a large deviation in one timeslot (in the case of a relatively large shipment) might be balanced by smaller deviations in two other timeslots. Consequently, the daily objective value decreases while the average WIP per timeslot becomes imbalanced. This analysis has provided us with a clear understanding of the development of the objective value and the WIP in relation to the number of added companies in the algorithm. Ultimately, these insights can be incorporated into the recommendations for the company.

6.4 Timeslot strategy scoring

Now that it is evident that the SA algorithm outperforms the current situation in terms of objective value and WIP, demonstrating validated performance, and considering that the number of companies primarily affects the objective value rather than the WIP, we will proceed to test the algorithm's selection of companies per iteration. To conduct this test, we will incrementally increase the number of selected companies by five. Since Company X currently does not have any agreements with companies, we will gradually build up the selection process.

Table 6.5: Timeslot strategies table comparison

Timeslot strategy	Companies [min, max]	Objective value	Average value	Difference previous strategy	Cumulative difference	Number of companies selected
Current	0	7673278	45674	0	0	0
A	[0,5]	6560006	39048	-6627	-6627	4
B	[5,10]	6461182	38459	-588	-7215	9
C	[10,15]	6342702	37754	-705	-7920	14
D	[15,20]	6310824	37564	-190	-8110	17
E	[20,25]	6228812	37076	-488	-8598	22
F	[25,30]	6075968	36166	-910	-9508	29

The above Table 6.5 presents the average differences among the various timeslot strategies. The current solution forms the upper bound, and the strategies are formed by incrementally increasing the minimum and maximum number of added companies by 5. The right column shows the number of companies selected by the algorithm. It is important to note that this is an example solution, and running the same algorithm with the same settings may yield different numbers of selected companies. In this case, assigning a fixed timeslot to 4 companies results in an improvement of 6,627 items, meaning an average daily deviation of 6,627 fewer items from the desired item flow. This improvement gradually increases and culminates in a difference of 9,508 items of average deviation per day.

Figure 6.10: Timeslot strategies score comparison

The objective values of the different timeslot strategies are plotted on a scoring line in Figure 6.10, and the purpose of Figure 6.10 is to demonstrate how all the different timeslot strategies perform compared to the current situation.

6.5 Conclusion

This chapter provides an answer to the question: "How does our algorithm perform?"

Firstly, we conducted an analysis of the baseline results by comparing the objective value and WIP development with the current situation. The average objective value decreased from 45,674 to 36,645 (-19.77%). This means that, on average, there are 9,029 fewer items that are either over or under-allocated per day. Comparing the baseline WIP per timeslot with the current WIP, we observed improvements ranging from -44.04% to -98.59%, indicating a significant enhancement.

The validation experiments demonstrated that the performance of our algorithm is valid as it outperforms all three predefined strategies. The average objective value of the random strategy is 4,396 items higher (+12%), the item-based strategy is 4,331 items higher (+11.82%), and the frequency-based strategy is 7,143 items higher (+19.49%). The results of the three predefined strategies also indicate that the model's output is highest when companies that deliver a large number of items per shipment are selected as input. The data showed that these companies often have a relatively high arrival frequency as well. Additionally, the development of the baseline WIP demonstrated an improvement compared to the three predefined strategies.

Through a sensitivity analysis, we demonstrated a clear trade-off between adding companies to the solution and reducing the objective value. The relatively small improvement observed by adding more than 30 companies does not outweigh the complexity of making agreements with all those companies.

Finally, we compared the current and six timeslot strategies with an incrementing number of companies assigned a fixed timeslot. The results showed that assigning timeslots to only four companies already led to an average objective value reduction of 6,627 items. A scoring line illustrated how close we approached the optimal situation. The reason for implementing a fixed timeslot approach was that the optimal situation is practically unattainable, whereas this timeslot method is feasible. Ultimately, the model significantly improved the objective value.

Chapter 7

Model implementation

This chapter provides recommendations in the form of guidelines and gives an answer to the question: "How should the timeslot indication model be implemented in practice?". The foundation of the implementation is depicted in figure 7.1, which illustrates the three iterative design phases:

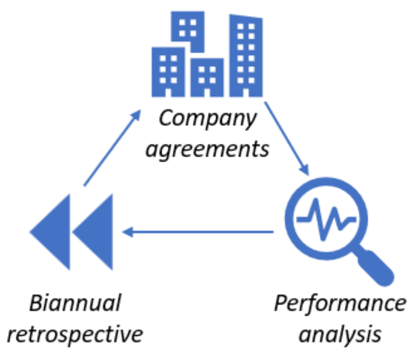

Figure 7.1: Implementation design phases

Company agreements

The first phase involves establishing timeslot agreements with suppliers. They are responsible for ensuring that the logistics party delivers their items to the Warehouse at the agreed moment in time. Our research has indicated that it is best to make agreements with companies that bring a relatively large number of items per shipment and have a high arrival frequency, as these factors are often correlated. As an incentive, we can promise suppliers that their logistics partners do not have to wait in line in front of our warehouse. Within our research, we encountered companies that consistently made significant errors regarding the reported number of items. Therefore, when selecting companies to allocate a fixed timeslot, it is important to consider the reliability of the reported item counts before making an agreement. Our research has shown that selecting four companies already yields a significant improvement in both the average number of item deviations.

Performance analysis

During the second phase of the implementation, retrospective analyses are conducted to evaluate the performance of the timeslot allocation. This can be done in a similar manner as in our research, where daily assessments are made of the incoming items and their corresponding timestamps from the previous day. The output of these analyses will ultimately be utilized in the third phase.

Biannual retrospective

The third phase consists of conducting a biannual retrospective meeting where the results of the analyses from the second phase are discussed. During this meeting, the performance of the objective value and WIP before, after, and during the new timeslot agreements is examined. If improvements are observed, it may be decided to add additional companies to the agreements. In case of a decrease, the current selection of companies can be reevaluated and adjustments can be made. When modifying the selection of companies, it is important to minimize changes to existing agreements. For example, if an additional company is added, it is preferable not to modify the timeslots assigned to the current companies. This helps avoid the unnecessary adjustment of multiple agreements, which would increase complexity. Ultimately, this third phase transitions back to the first phase, initiating a new iteration.

Lessons learned historical experience

In the past, Company X has previously attempted to implement a timeslot planning system and valuable lessons have been learned from that experience. It is crucial to have clear descriptions in the contract regarding the expectations for drivers upon arrival at the warehouse. Drivers should be aware of the right paperwork needed in case they have to show it and the availability of the fast lane. Additionally, allowances should be made for the possibility of drivers arriving late, and it should be acknowledged that the reported quantity of items may not always match the actual received quantity. Therefore, when conducting the analysis, we recommend taking into account the received quantity of items.

Chapter 8

Conclusions & Recommendations

This final chapter provides an answer to our main research question: "How can long-term inbound shipment planning be used to reduce the amount of daily WIP at the inbound area of the Company X's Warehouse 1?". Section 8.2 explains the limitations of our model and highlights important considerations to keep in mind when interpreting the results. Additionally, section 8.3 provides recommendations for both Company X and future research. Finally, in section 8.4, we elaborate on the contribution of our research to the existing literature and its practical implications within Company X.

8.1 Conclusion

During this research, we developed a Simulated Annealing heuristic that analyzes the impact of assigning a fixed timeslot to a set of retailers on the incoming flow of items, using historical data.

The motivation for this research stems from the high WIP quantity observed between the unloading and receiving phases of the inbound process. This leads to challenges such as the need to reject shipments due to space constraints, failure to meet the promised dock-to-stock time of 72 hours, and the utilization of floor space that does not positively contribute to the process.

A literature study was conducted as a foundation for developing our Mixed-Integer Linear Programming model and the Simulated Annealing algorithm. It also provided characteristics of our inbound truck scheduling problem and defined it as an offline identical parallel machine scheduling problem. Previous studies showed that our scheduling problem is prevalent within the context of cross-docking operations. Because of the size of our problem, we decided to develop a Simulated Annealing algorithm to approximate the solution.

The first phase of the model imports historical data and uses a Simulated Annealing algorithm to allocate fixed timeslots to a number of companies, aiming to minimize the total deviation between the desired item flow and the actual item flow on a daily basis. Various operators are utilized to adjust the timeslot for each company and to add or remove companies from the solution. The second phase evaluates a determined timeslot strategy by comparing the objective value with the current and optimal values. The current value is obtained by making no adjustments to the historical data and calculating the objective value.

Numerical experiments have shown that providing a fixed timeslot to companies that deliver a relatively large number of items per shipment leads to a decrease in the objective value and an improvement in the development of the average daily WIP. The baseline settings resulted in an average objective value decrease from 45,674 to 36,645 items (-19.77%), with 27 companies assigned a fixed timeslot. The WIP per timeslot decreased by percentages ranging from -44.04% to -98.59%. The Simulated Annealing algorithm outperforms the random strategy, the item-based strategy, and the frequency-based strategy, with average objective values of 4,396 items (+12%), 4,331 items higher (+11.82%), and 7,143 items higher (19.49%), respectively. Sensitivity analysis showed that adding a large number of companies at once does not outweigh the complexity associated with making all those agreements. The objective value increases relatively less, and the development of the WIP does not become more consistent as a result of our model.

Our research has demonstrated that providing a fixed long-term timeslot to a number of companies results in a reduction in the amount of WIP at the inbound area of the Company X Fulfillment Center 1. Just allocating a fixed timeslot to four companies already leads to an average objective value reduction of 6,627 items, meaning that there are 6,627 items that are either over-allocated or under-allocated per day. For the implementation of our methods, an iterative process can be utilized, consisting of making company agreements, analyzing the results, and reconsidering the timeslot allocation agreements.

8.2 Limitations

The limitations of our model are mainly explained by the assumptions made to simplify reality. We assume that shipments arrive all at once, while in reality, they may arrive dispersed over multiple trucks. Conversely, multiple individual shipments could be consolidated into the same truck. Additionally, we assume a standard unloading processing time of 10 minutes, with an additional minute per extra pallet. Furthermore, we do not account for the additional time it takes when the truck arrives at the gatekeeper. This can include the time needed for security checks, paperwork verification, and other administrative procedures before the unloading process can begin. Moreover, shipments cannot arrive earlier or later than the defined timeslots, which makes it challenging for the model to fill the first timeslot and causes shipments that fall outside the last timeslot to be included in the last timeslot. Lastly, this model results in a reduction in the amount of WIP, which automatically leads to an improved daily WIP compared to the previous situation. However, the effect of the number of added companies is not visible. In reality, as more companies arrive at a fixed time, the process may become more regulated.

8.3 Recommendations for future research

We recommend that Company X conducts further research into the exact process times of the entire arrival process. This includes analyzing the processing time at the gatekeeper, the time it takes to drive to the dock, the docking procedure, the unloading of the pallets, and finally, the undocking process. By obtaining accurate data on these process times, a more realistic estimation can be made regarding allocating a timeslot to a company.

Additionally, it is important to take into account multi-site deliveries. It is possible for a truck to have a shipment destined for multiple warehouses. For instance, Warehouse 1 consists of two sub-warehouses located in the same building but have separate docks. Therefore, the long-term timeslot planning for Warehouse 1 should be coordinated with both warehouse parts to ensure that a company is not scheduled to arrive at the first part in the morning and then at the second in the afternoon. Instead, the truck should be able to visit both docks in a single trip.

Finally, we recommend conducting further research to optimize the WIP development. The objective of our model is the reduce the number of items in the WIP, and although it improves the current situation, it does not necessarily mean that the number of items arriving at the warehouse is the same at every time interval.

8.4 Contributions to literature and practice

Within our research, we have developed a Simulated Annealing algorithm that is capable of reducing the number of items in the WIP area between the unloading - and receiving phases. This reduction is achieved by assigning fixed delivery timeslots to a number of companies. The objective function central to this algorithm minimizes the summation of the absolute difference between the preferred number of incoming items and the actual number of incoming items. Previous research within the field of cross-docking operations showed us the possibility of implementing a Simulated Annealing algorithm to solve similar inbound truck scheduling problems. Our research adds a theoretical contribution to these studies. Also, the combination of a Simulated Annealing algorithm and the allocation of fixed timeslots to suppliers to reduce the number of items idle between the unloading - and receiving phases has not been observed in the current literature.

The primary contribution to practice is the different perspective we have introduced toward the impact of timeslot agreements. Prior to this research, Company X perceived the solution of making agreements with its suppliers to be too complex, because of the large number of suppliers they work with. We have demonstrated that a minor adjustment, namely the implementation of fixed time slots for four companies, already yields a significant positive impact. With this insight, Company X can take the first steps in reducing the number of items in the WIP area by starting to make contact with the first group of suppliers.

Bibliography

[1] S. Balin. "Non-identical parallel machine scheduling using genetic algorithm". In: *Expert Systems with Applications* (2011), pp. 6814–6821.

[2] D. Bertsimas and J. Tsitsiklis. "Simulated Annealing". In: *Statistical Science* (1993), pp. 10–15.

[3] C Blum and A Roli. "Metaheuristics in combinatorial optimization: overview and conceptual comparison". In: *Journal of Advanced Transportation* (2003), pp. 268–308.

[4] N. Boysen, D. Briskorn, and M. Tschöke. "Truck scheduling in cross-docking terminals with fixed outbound departures". In: *OR Spectrum* (2012), pp. 479–504.

[5] D. Buakum and W. Wisittipanich. "A literature review and further research direction in cross-docking". In: *Proceedings of the International Conference on Industrial Engineering and Operations Management* (2019), pp. 471–481.

[6] T. Chaari et al. "Scheduling under uncertainty: survey and research directions". In: *International Conference on Advanced Logistics and Transport* (2014), pp. 229–234.

[7] C. Floudas and X. Lin. "Mixed Integer Linear Programming in Process Scheduling: Modeling, Algorithms, and Applications". In: *Annals of Operations Research 139* (2005), pp. 131–162.

[8] R. Haupt. "A Survey of Priority Rule-Based Scheduling". In: *Lehrstuhl fiir Personalwirtschaft* (1988), pp. 3–16.

[9] C Kirkpatrick et al. "Optimization by Simulated Annealing". In: *Science* (1983), pp. 671–680.

[10] A. Ladier and G. Alpan. "Cross-docking operations: Current research versus industry practice". In: *The International Journal of Management Science* (2015), pp. 1–21.

[11] M. Madani-Isfahani, R Tavakkoli-Moghaddam, and B Naderi. "Multiple cross-docks scheduling using two meta-heuristic algorithms". In: *Computers Industrial Engineering* (2014), pp. 129–138.

[12] E. Mokotoff. "An exact algorithm for the identical parallel machine scheduling problem". In: *European Journal of Operations Research* (2004), pp. 758–769.

[13] Michael L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Fifth edition. Springer International Publishing AG Switzerland, 2016. ISBN: 978-3-319-26578-0.

[14] D. J. Rader and Rader Jr. *Deterministic Operations Research: Models and Methods in Linear Optimization*. John Wiley Sons, INC., 2010. ISBN: 978-0-470-48451-7.

[15] G. Tadumadze et al. "Integrated truck and workforce scheduling to accelerate the unloading of trucks". In: *European Journal of Operational Research* (2019), pp. 343–362.

[16] P. Wolff, J. Huo, and H. Pfohl. "Mixed-Integer Programming formulations for the truck scheduling problem with fixed outbound departures". In: *Scientific Journal of Logistics 17* (2021), pp. 243–251.

[17] M. Yazdani, B Naderi, and M. Mousakhani. "A model and metaheuristic for truck scheduling in multi-door cross-dock problems". In: *Intelligent Automation Soft Computing* (2015).

[18] W. Yu and P. Egbelu. "Scheduling of inbound and outbound trucks in cross docking systems with temporary storage". In: *European journal of Operational Research* (2006), pp. 1–20.

Appendix A

Truck Arrivals per day

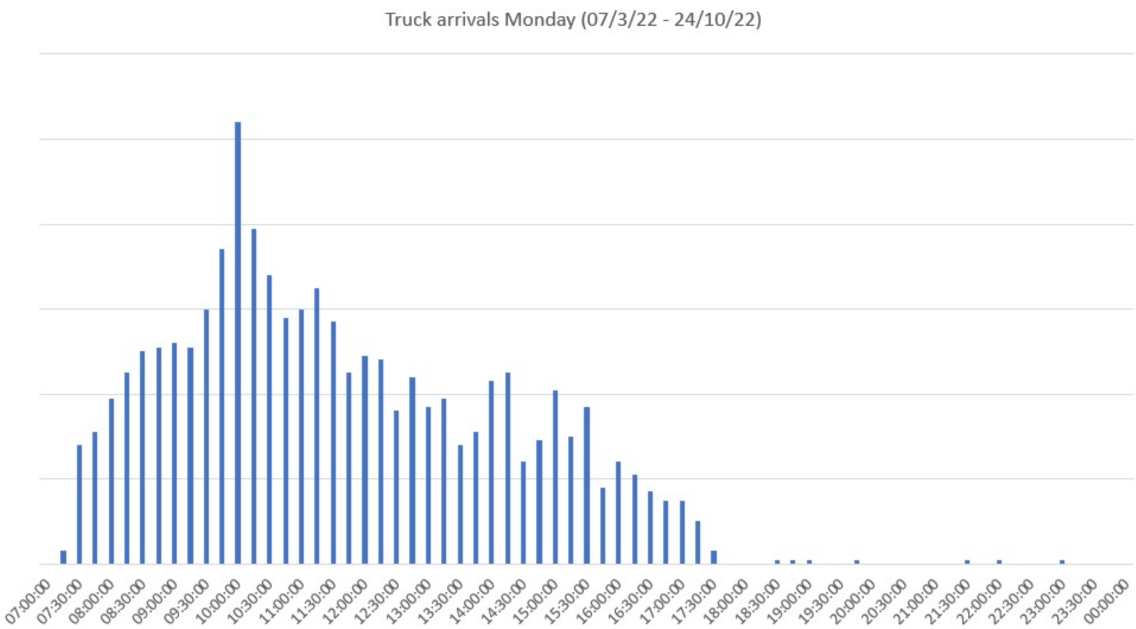

Figure A.1: Truck arrivals Monday

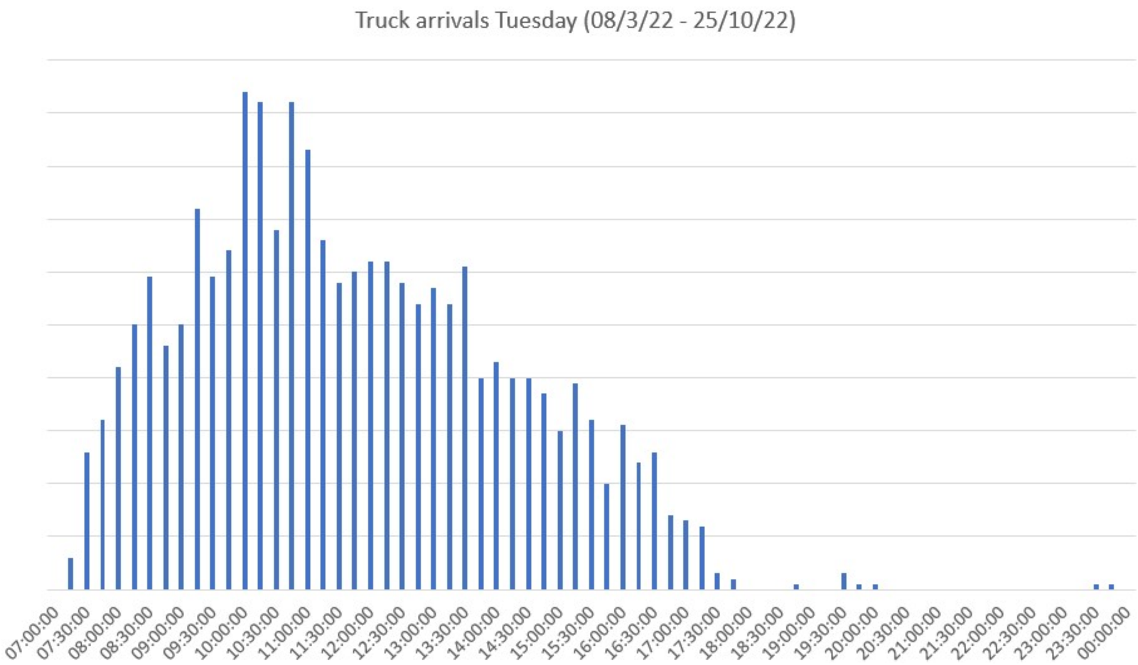

Figure A.2: Truck arrivals Tuesday

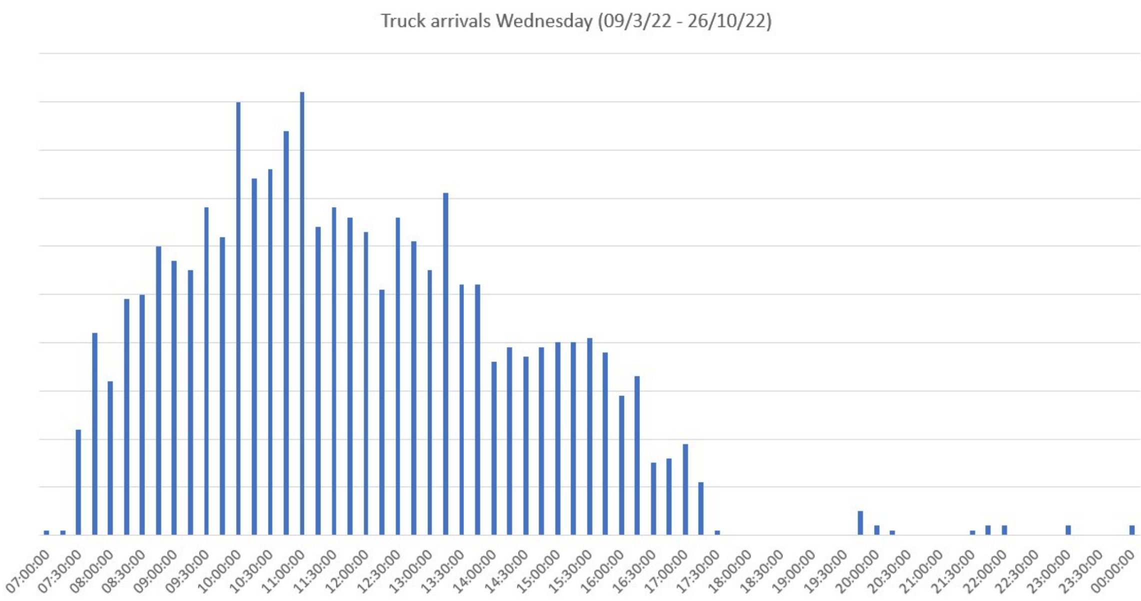

Figure A.3: Truck arrivals Wednesday

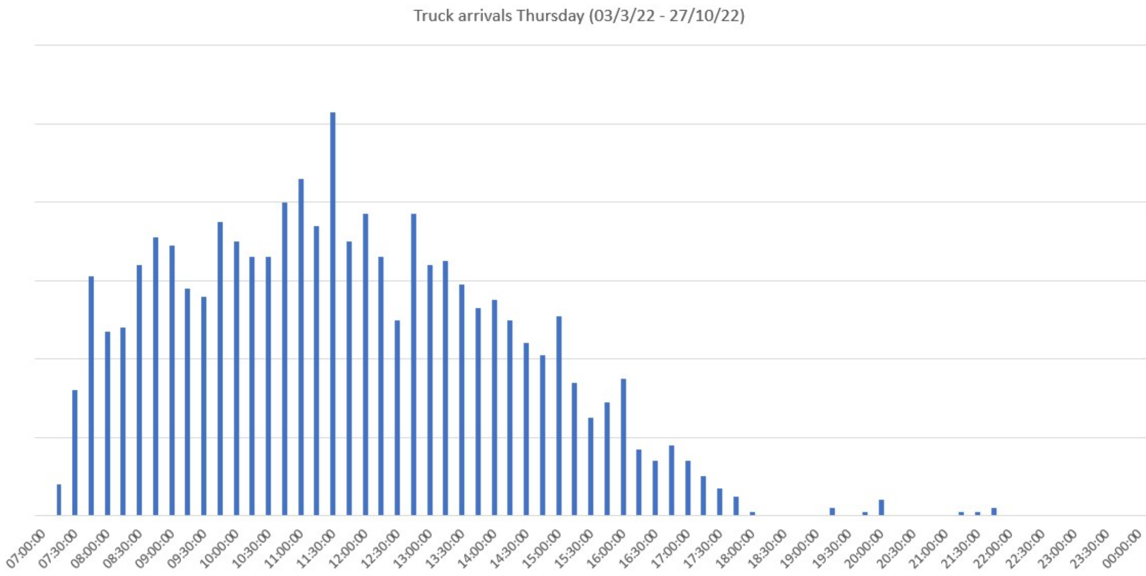

Figure A.4: Truck arrivals Thursday

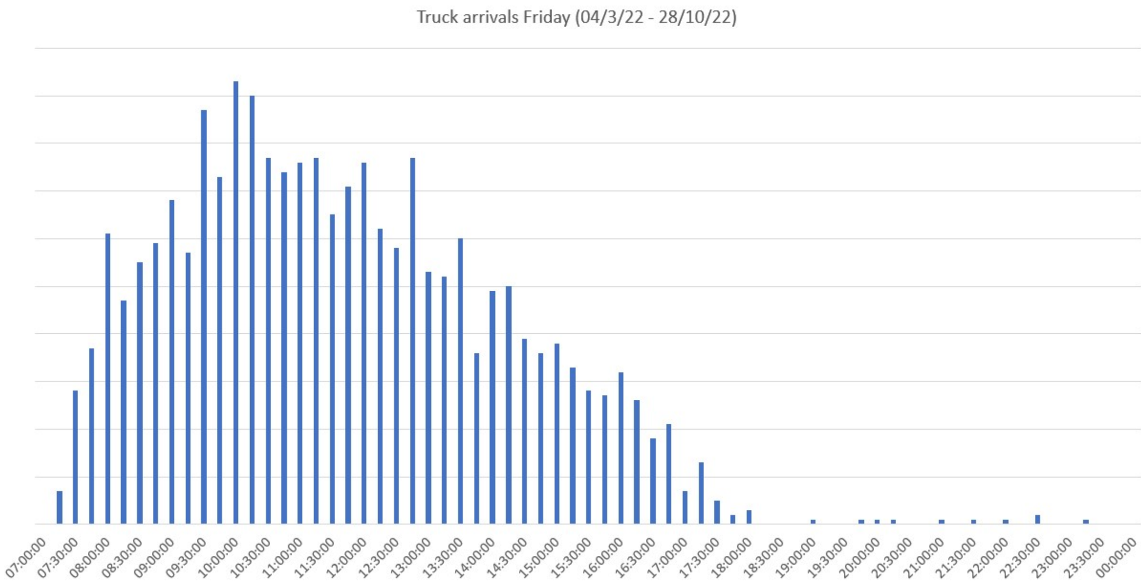

Figure A.5: Truck arrivals Friday

Appendix B

Simulated Annealing operator algorithms

Algorithm 4: Swap

Function Swap(Schedule, AdjustableCompanyList):

$RandomCompany1 \leftarrow random.choice(AdjustableCompanyList)$;
$RandomCompany2 \leftarrow random.choice(AdjustableCompanyList)$;
while $RandomCompany1 = RandomCompany2$ **do**
    $RandomCompany2 \leftarrow random.choice(AdjustableCompanyList)$;
end
$TempInterval \leftarrow$ *Arrival interval value* **of** $RandomCompany1$ **from** $Schedule$;
$SwapInterval \leftarrow$ *Arrival interval value* **of** $RandomCompany2$ **from** $Schedule$;
Arrival interval value **of** $RandomCompany1$ **from** $Schedule \leftarrow SwapInterval$; *Arrival interval value* **of** $RandomCompany2$ **from** $Schedule \leftarrow TempInterval$;

Return *Schedule*

Algorithm 5: Move

Function Move(Schedule, AdjustableCompanyList):

$RandomCompany \leftarrow random.choice(AdjustableCompanyList)$;
$RandomNumber \leftarrow random.uniform(0, 1)$;

if $RandomNumber \leq 0.5$ **then**
    $NewInterval \leftarrow$ *Arrival interval value* **of** $RandomCompany$ **from** $Schedule$;
else
    $NewInterval \leftarrow$ *Arrival interval value* **of** $RandomCompany$ **from** $Schedule$;
end

if $NewInterval < 1$ **then**
    $NewInterval \leftarrow 2$;
else if $NewInterval > 11$ **then**
    $NewInterval \leftarrow 10$;
else
    Arrival interval value **of** $RandomCompany$ **from** $Schedule \leftarrow NewInterval$;
end

Return *Schedule*

Algorithm 6: Insertion

Function Insertion(Schedule, AdjustableCompanyList):

$RandomCompany \leftarrow random.choice(AdjustableCompanyList)$;
$RandomInterval \leftarrow random.int(1, 11)$;
Arrival interval value **of** *RandomCompany* **from** *Schedule* $\leftarrow$ *RandomInterval* ;

Return *Schedule*

Algorithm 7: RandomLocalSearch

Function RandomLocalSearch(Schedule, AdjustableCompanyList):

```
BestIntervalValue ← CalculateObectiveValue(Schedule)
  RandomCompany ← random.choice(AdjustableCompanyList);
RandomInterval ← random.int(1, 11);
for t = 1 to T do
    PotentialSchedule ← Schedule.copy();
    Arrival interval value of RandomCompany from PotentialSchedule ← t + 1;
    PotentialIntervalValue ← CalculateObjectiveValue(PotentialSchedule);
    if PotentialIntervalValue < BestIntervalValue then
        BestIntervalValue ← PotentialIntervalValue;
        BestInterval ← t + 1 ;
    end
end
Arrival interval value of RandomCompany from Schedule ← BestInterval;
Return Schedule
```

Algorithm 8: AddCompanyToList

Function AddCompanyToList(Schedule, AdjustableCompanyList, PotentialCompanyList):

```
RandomCompany ← random.choice(PotentialCompanyList);
while RandomCompany in AdjustableCompanyList do
    RandomCompany ← random.choice(PotentialCompanyList);
end
AdjustableCompanyList.append(RandomCompany)
Arrival interval value of RandomCompany from Schedule ← random.int(1, 11);
Return Schedule, AdjustableCompanyList
```

Algorithm 9: RemoveCompanyFromList

Function AddCompanyToList(Schedule, ScenarioData, AdjustableCompanyList, PotentialCompanyList):

```
RandomCompany ← random.choice(AdjustableCompanyList);
AdjustableCompanyList.remove(RandomCompany);
Arrival interval value of RandomCompany from Schedule ← Arrival interval value of
  RandomCompany from ScenarioData;;
Return Schedule, AdjustableCompanyList
```

Algorithm 10: SwapCompanies

Function SwapCompanies(Schedule, ScenarioData, AdjustableCompanyList, PotentialCompanyList):

```
RandomCompany1 ← random.choice(AdjustableCompanyList);
RandomCompany2 ← random.choice(PotentialCompanyList);
while RandomCompany1 = RandomCompany2 or RandomCompany2 in AdjustableCompanyList do
    RandomCompany2 ← random.choice(PotentialCompanyList);
end
AdjustableCompanyList.append(RandomCompany2);
Arrival interval value of RandomCompany2 from Schedule ← Arrival interval value of
  RandomCompany1 from Schedule;
AdjustableCompanyList.remove(RandomCompany1);
Arrival interval value of RandomCompany1 from Schedule ← Arrival interval value of
  RandomCompany1 from ScenarioData;
Return Schedule, AdjustableCompanyList
```

Appendix C

WIP development: number of companies

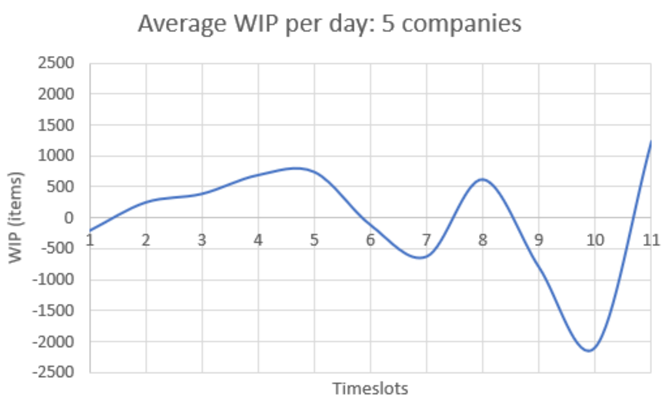

Figure C.1: Wip development 5 days

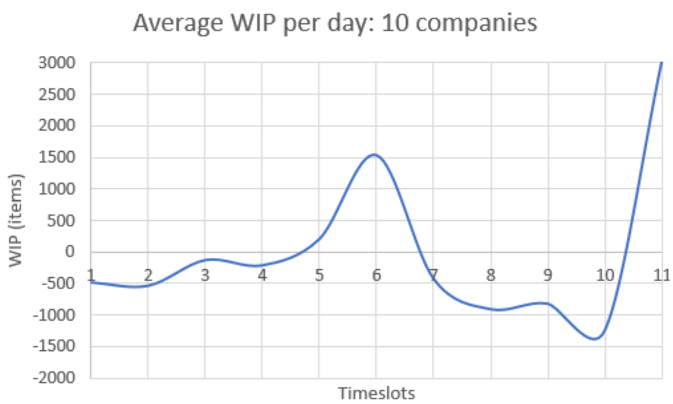

Figure C.2: Wip development 10 days

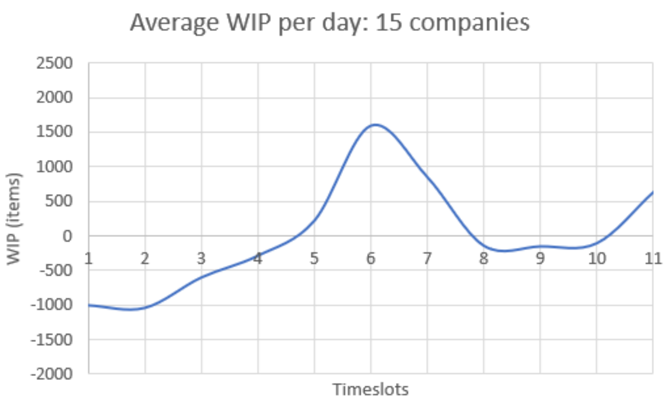

Figure C.3: Wip development 15 days

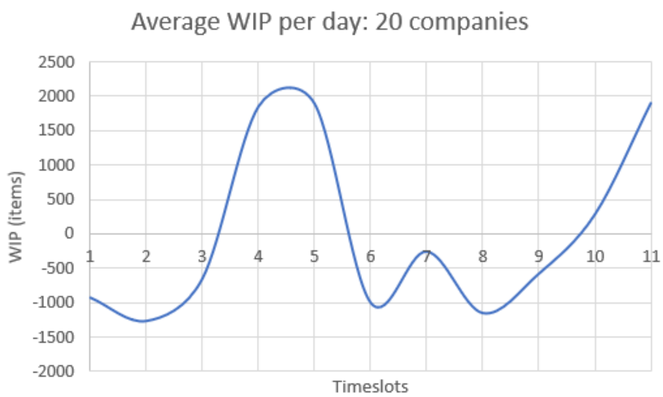

Figure C.4: Wip development 20 days

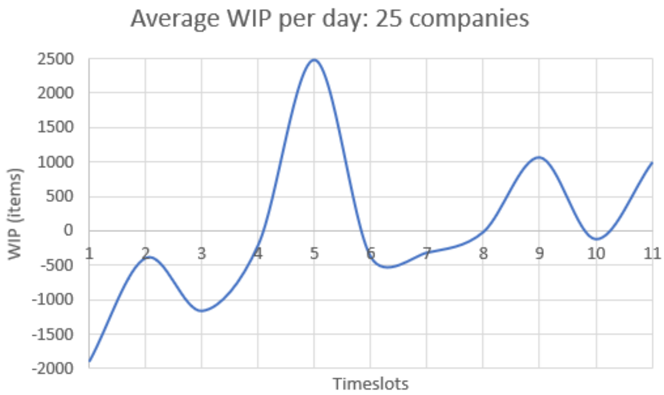

Figure C.5: Wip development 25 days

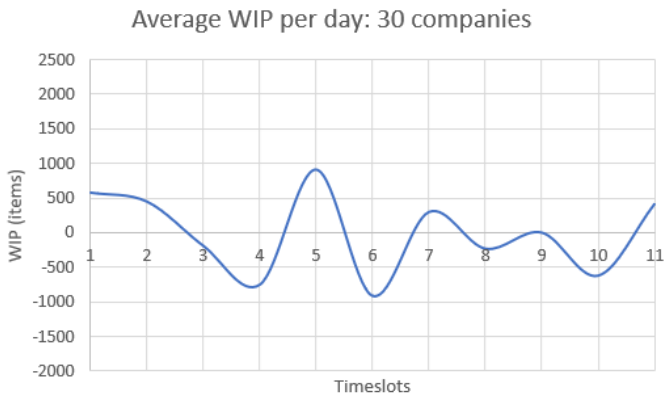

Figure C.6: Wip development 30 days

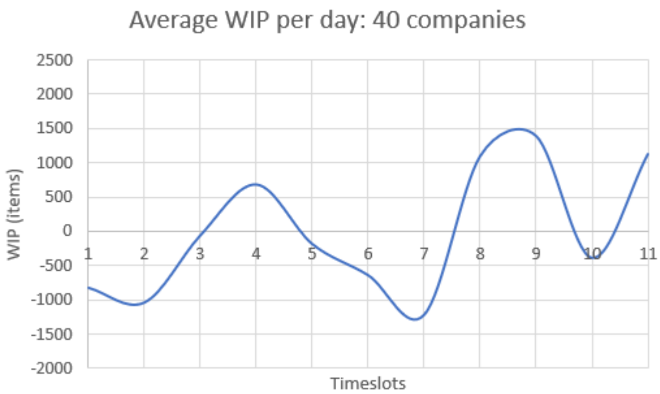

Figure C.7: Wip development 40 days

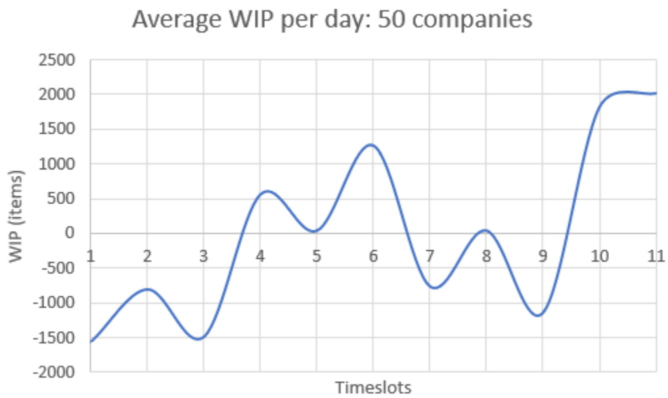

Figure C.8: Wip development 50 days

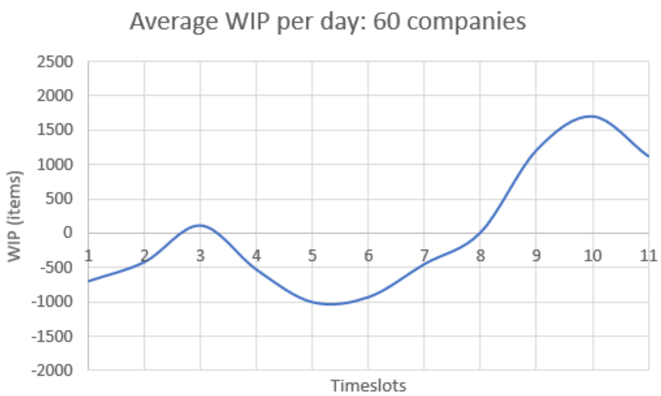

Figure C.9: Wip development 60 days