MASTER

Improving the delivery reliability and the makespan in a hybrid flowshop manufacturing environment

Jeuken, Susan

Award date:
2023

Thesis

Improving the delivery reliability and the makespan in a hybrid flowshop manufacturing environment

S. Jeuken, 1623389

In partial fulfillment of the MSc. Operations Management and Logistics

Supervisors
Dr. ir. N.P. Dellaert
Dr. L. Tan
T. van Dijk

Eindhoven, April 11, 2023

Eindhoven University of Technology
School of Industrial Engineering & Innovation Sciences
Series Master Theses Operations Management And Logistics

Key words:
Hybrid flowshop scheduling problem, Dispatching rules, Simulated Annealing, Product sequencing, Batch production, Improving delivery reliability, Improving makespan

Abstract

Timely deliveries of materials used for a building project is essential for finishing a building project on time. Therefore, a research has been conducted at the manufacturer of lacquered interior doors with the goal of improving the delivery reliability and the makespan by changing the sequencing method. The manufacturing environment in which this research has been conducted is a hybrid flowshop: several production stages can be distinguished of which some operate in parallel, and products may skip stages. The products start flowing individually and are grouped together in batches after the initial stages. Several factors are included in the scope of the research, namely: failures and maintenance, rework, various priority levels, limited buffer sizes, and overwork. A mathematical model is developed for the sequencing problem, but the model needs too much computation time to be a realistic option for improving: a new solution is needed before the previous solution is obtained. Therefore, three (meta)heuristics are tested by means of a simulation model, namely the Earliest Due Date (EDD) rule, the RR-rule, and the Simulated Annealing (SA) metaheuristic, which used an EDD-rule per operation as initial sequence. For this EDD-rule per operation a due date per operations was determined based on the remaining service times, slack, and average waiting times. Based on the results obtained by the simulation model, the SA metaheuristic generated the best results, but these results were comparable to the results generated with the initial solution used for the SA metaheuristic, which were obtained by the EDD per operation rule. As the EDD-rule per operation needed far less computation time, this heuristic is considered optimal, which increased the delivery reliability and makespan considerably.

Preface

This Master thesis marks the end of my journey at the Eindhoven University of Technology (TU/e). Most of my time at the TU/e was during the Covid-19 pandemic, which challenged my time here significantly, but also provided opportunities. In October 2022 I started writing my thesis at Berkvens, an interesting company in my opinion, where I felt welcome immediately. This interest was mutual as I was asked to apply for the vacancy for Operations Engineer, a position that I will start working in coming October.

I would like to thank the people that helped me during this project. First and foremost, I want to thank my mentor Nico Dellaert for his help and guidance during this project, but also during the rest of my time at the TU/e. Nico helped me wherever he could and always responded to my mails promptly, gave me critical feedback, and supported me throughout the project. Secondly, I would like to thank my second supervisor Lijia Tan for her help and feedback during the project.

At Berkvens, I would like to thank Tom van Dijk for his guidence, support, and help during my time at Berkvens. He gave me many insights in the working of the factory and directed me to other relevant people that could help me with my research. Furthermore, I would like to thank all the operators at the LDF, as well as the foremen and people at the business office, and all the other people that helped me in this research.

Lastly, I want to thank my family and friends. Without their support and patience and believe in my abilities I would have been lost. I appreciate all the space and understanding they have shown me and thank them for reminding me from time to time that life entails more than just working.

Susan Jeuken

Someren, The Netherlands, April 7, 2023

Management summary

The research in this thesis is executed in a company named Berkvens B.V., Berkvens in short. Berkvens is a manufacturer of interior doors and door frames, based in Someren, The Netherlands. The Lacquered door factory (LDF) of Berkvens, the environment in which the research is executed, is a hybrid flowshop (HFS) environment, which is defined as follows: "A HFS consists of series of production stages, each of which has several machines operating in parallel. Some stages may have only one machine, but at least one stage must have multiple machines." ([Linn and Zhang, 1999], p. 57). The LDF consists out of several production stages: first products are sawed, then labelled, pressed, edge processed, drilled, CNC, painted, special (SPEC) features are added, and lastly the products are packed. Not all products need to flow through all stages; some stages such as the CNC and the SPEC operations may be skipped. In the first stages of the process, the sawing and labelling operations, the products move individually. From the presses onward, the products move in batches. The sales orders can be classified into two main types, namely made-to-order (MTO) products and made-to-stock (MTS) products. In general, production of MTS orders is easier because these orders are more standardized. MTO products can be standardized as well, but MTO doors can also have a high degree of specialization. The customer for the MTS orders is the Berkvens warehouse, whereas the customer for MTO orders is an external company, which means that late deliveries of MTO products have more impact when compared to late deliveries of MTS products.

Problem description

Currently, the lacquered door factory (LDF) of Berkvens struggles with maintaining a high delivery reliability. The LDF works with two due dates, an internal one and an external one, where the external due date is set two days after the internal due date. The key performance indicator (KPI) for the internal due date, the one that is most important for the LDF, is set to 98%. The measured performance on the service level of the LDF is only 88.5%, which is considerably lower than the goal of 98%. This realized performance results in a decreased performance on the external delivery reliability as well, which was 96.4% in the measured period, against a KPI of 99.5%.

Placing the interior doors is one of the last steps in building homes, thus Berkvens operates in the last stages of the building process. This means that disruptions in the processes that preceded the placement of the interior doors influence this step, and shifts in due dates occur frequently. Currently, the production orders for the doors produced in the LDF are released several weeks before the doors' due date. This could result in an released order for which the due date might be shifted back later. When the makespan of products is decreased, the order could be released later, and more advantage could be taken from shifting due dates.

The above two problems lead to the main research question of this thesis, which is: *How to improve the service level and the makespan of doors produced in the lacquered door factory of Berkvens?* Several factors will be researched to answer this research question, namely the parameters and performance of the individual operations and the interaction between the operations, the differences between MTO and MTS orders and how to deal with those differences, the process of determining the due date and what adjustments can be made in this process, and the process of determining the sequence of jobs and how this process can be altered to improve the service level and the makespan of products.

Methodology

The factors mentioned above have been researched by a series of methods. Data-analysis has been used as well as interviews to determine the current state of processes. Literature research has been done to determine which model can be used best for the sequencing and how to design these models. Furthermore, a mathematical model as well as a simulation model is developed for testing the designed models. Lastly, the designs have been compared and an implementation plan is made for the best design. Some more details on methodology are given below.

The first factor that is researched are the parameters and current performance of the operations in the LDF and the interaction between the operations. This is done by gathering data about the operations and products and analyzing this data. Inconsistencies in the data are noticed and in-depth investigation about those inconsistencies is done. Interdependencies are investigated by examining the routes of the products and identifying main routes and remaining transitions. The performance of the operations is evaluated by comparing the arrival- and service rates of all operations to be able to determine a service rate.

The second and third factors that are researched are the differences between MTO and MTS orders and the process of determining the due date, respectively. The second factor is researched based on a combination of data analysis and interviews. The third factor is researched with interviews with the head of Berkvens' sales department.

The fourth factor that is researched is the current sequencing method and how this sequencing method could be improved to improve the makespan and service level. This is researched based on the results of the first three factors and some additional investigations. The current sequencing method is investigated by conducting interviews, and several methods for improving the sequencing have been determined by performing literature reviews and are tested with a simulation model. The results of the various methods are compared and the best one is chosen as final model.

Results

Based on the results of the data-analysis on the parameters and performance of the operations, the parameters are determined an have been compared with the characteristics of the demand process. The conclusion is that the operations in the LDF should be able to deal with the demand, as the utilization rates of all operations are between 0.7 and 0.9.

The research on the analysis on the differences between MTO and MTS products concluded that MTO orders make up approximately 70% of the total orders and have a shorter lead time, in general between 4 and 9 days unless an external operation is needed, compared to a standard lead time of 20 days for MTS orders. An analysis of the individual arrival processes for MTO and MTS orders yielded no significant results, so no distinction can be made between those two and no claims can be made about differences in order sizes or order patterns. It can however be concluded that the MTO orders should have a higher priority than MTS orders, as the consequences of late deliveries of MTO orders are more extensive than the consequences of late deliveries of MTS orders, due to safety stocks for MTS orders.

Investigation on the process of determining the due date resulted in an increased insight in the processing of orders. Normally, the processing of an order is started three weeks after the order is received, and the due date is 8 weeks after the order is received. Berkvens operates in the building industry, which is known to be prone to disruptions. Placing the interior doors and door frames happens in one of the last stages of building new houses, which means that many disruptions in the building process influence Berkvens' process. As a result, shifts in due dates occur frequently. However, the sales process is a complex process and out of scope for this research, hence adjusting the process of determining the due date will not be done.

The current sequencing method has been investigated. Currently, batches and products have a priority level based on the type of order and its characteristics. Operators should adhere the priority rules, but can otherwise determine which product or batch to process next themselves. Based on simulation of this process, the average service level is 90.8%, the average service level for MTO orders is 88.1%, and the average makespan is 74.4 hours.

Three sequencing (meta)heuristics have been selected based on literature investigations and have been designed and tested for the LDF of Berkvens. The first heuristic is the Earliest Due Date (EDD) dispatching rule, which sequencing the products and batches according to their due dates, where the product or batch with the due date coming up first is processed first. The second heuristic is the RR dispatching rule, which prioritizes products based on the remaining service time, the slack, and the average waiting time. The third metaheuristic is the Simulated Annealing, which determines the sequence based by testing various solutions and selecting the best one. This Simulated Annealing metaheuristic uses a sequence as input, and this initial sequence is based on an EDD rule per operation. A due date per operation has been determined by taking several factors into account, including the waiting times, remaining times, and characteristics of the order type. All three metaheuristics as well as the EDD rule per operation have been tested with the simulation model. Based on the results it can be concluded that the SA metaheuristic had the best performance, closely followed by the EDD rule per operation. As the performance of these two metaheuristics was very similar and the SA rule took almost six times as long to compute, the EDD rule per operations is chosen as the most optimal solution. The average service level will increase to 94.6%, the average service level of MTO orders to 91.6%, and the average makespan will decrease to 67.42 hours.

Recommendations

Before the new sequencing method can be implemented, a couple of steps need to be done. First of all, support for the new model and understanding of the model should be created, which can be done by gathering all stakeholders and providing information on why the new method is necessary, what the new method entails, and how it improves the performance. Furthermore, the new heuristic needs information on the product and batch type to be able to determine a due date per operation. When Berkvens' new ERP system is implemented in 2024, gathering this information should be easy and determining the deadlines can be done by the system. Lastly, some facilities should be added or adjusted to be able to communicate the new sequence to the operators clearly.

Besides the recommendations for the implementation of the new sequencing heuristic, several other recommendations for research directions to improve the the performance of the LDF even further can be done. First of all, the sales process can be analyzed and based on the results of the analysis, improvements might be made on the determination of the due date. This could improve the performance of the LDF and the other production facilities of Berkvens, as disruptions in the process could mean an advantage for Berkvens as well. Furthermore, the cause of disruptions in various processes could be investigated and hopefully diminished, leading to less disruptions and thus a smoother flow. Besides that, a research project could be executed on the amount and cause of rework, with the goal of reducing the effect of rework on the process. Lastly, more insight in the system would increase the ability to carry out reliable data-analyses, which in turn would lead to more sounds conclusions. Therefore, the quality and amount of data should be improved.

Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	**Meaning**
Avg	Average
CI	Confidence Interval
DES	Discrete Event Simulation
EDD	Earliest Due Date
EP	Edge Processing
ERP	Enterprise Resource Planning
FES	Future Event Set
GA	Genetic Algorithm
HFS	Hybrid Flow Shop
HPL	High Pressure Laminate
LDF	Lacquered Door Factory
Loc	Location
MP	Mass Press
MTO	Make-to-Order
MTS	Make-to-Stock
NLC	New Logistic Concept
PO	Production Order
SA	Simulated Annealing
SPEC	Special features
SPT	Shortest Processing Time
SRQ	Sub Research Question
SSP	Small Series Press
TS	Tabu Search

1 Introduction

The world population is growing, and with it the need for houses. In 2022 over 74 thousand new construction homes were built in The Netherlands alone [CBS and Swagerman, 2023], and this number of new homes has been this high over the last 7 years. This high demand for houses led to increased pressure on contractors, which in turn increased the need for suppliers to be reliable, such that projects can be finished within the agreed makespan. Suppliers should strive for the on-time-delivery of products to be as high as possible, such that the whole building project will be completed in time. After all, a whole construction project could get delayed when one supplier fails to deliver their products on time.

One of those suppliers is Berkvens B.V., a manufacturer of interior doors and door frames. Berkvens has struggled with maintaining the delivery reliability as high as they would like it to be. The Lacquered Door Factory (LDF) of Berkvens, where the most common types of interior doors are produced, had a delivery reliability of 96.4% last year, while their Key Performance Indicator (KPI) stated a goal of 99.5%. This gap between actual and desired delivery reliability results in decreased customer satisfaction, penalties, and eventually a decrease in the number of customers. To avoid these consequences in the future, they want to increase their delivery reliability and ensure that the KPI is met.

Improving the delivery reliability is one of the main topics in the field of industrial engineering, as this is widely regarded as one of the highest priorities of companies. Much research has been conducted on how to improve the delivery reliability [Jeuken, 2022]. The specific situation in a company is the main factor that determines the best way to improve the delivery reliability, as well as the ability of companies to adapt certain parts of their operations, and the nature of the industry in which they operate.

The research conducted in this thesis focuses on increasing the delivery reliability of the LDF of Berkvens, which operation can be classified as a Hybrid Flow Shop (HFS), which is defined as: "A HFS consists of series of production stages, each of which has several machines operating in parallel. Some stages may have only one machine, but at least one stage must have multiple machines." ([Linn and Zhang, 1999], p. 57). A literature review has been conducted on increasing the delivery reliability in a HFS environment, but none of the articles reviewed had the same parameters as the situation encountered at Berkvens. This thesis has thus both theoretical value as well as practical value.

This thesis starts with a more detailed company and problem description in Chapter 2, which leads to the research design presented in Chapter 3. In Chapter 4 the current performance of the LDF is investigated, while in Chapter 5 the current sequencing and planning procedure is investigated. Then, a mathematical model for the sequencing problem is developed in Chapter 6. In Chapter 7 a simulation model is introduced. Next, in Chapter 8 several heuristics and metaheuristics are introduced for the sequencing problem, which are evaluated in Chapter 9. Lastly, in Chapter 10 the research question of this thesis is answered, an implementation plan is presented, and the results are discussed.

2 Company and problem description

In this section briefly describes the company in which the thesis is executed. The processes and further details will be covered later in the report. The problem the company encounters is described as well. The research design described in the next section is based on this problem description.

2.1 Company description

This thesis is executed with the company Berkvens B.V., Berkvens in short, which is located in Someren, The Netherlands. Berkvens is a manufacturer of interior doors and door frames for residential and commercial construction. The company is founded in 1933 and celebrates its 90th birthday in 2023. Berkvens is one of the companies in the XiDoor holding, which is the overarching organization that entails three other companies, Svedex, Bod'or, and Dextura, which are also door manufacturers. Besides the presence on the Dutch market, XiDoor also operates in France, England, and Germany.

Berkvens has approximately 480 employees and had a revenue of 95 million euros in 2021. Its facility consists out of an office and four factories: the lacquered door factory (LDF), High Pressure Laminate (HPL) doors factory, wooden door frame production, and steel door frame production. Approximately 600,000 doors and 400,000 door frames are produced annually, which are used to serve the Dutch, English, and French market.

This thesis research takes place in the lacquered door factory (LDF). Various door configurations are produced here, and some doors are made-to-stock (MTS) while others are made-to-order (MTO). There are five main types of doors: A doors are MTS for the building of houses, AA doors are MTS for trade, B doors are anonymously MTO for the building of houses, BB doors are anonymously MTO for trade, and C doors are MTO for projects. Anonymously MTO means that the production of the door starts when an order arrives, but that the door is not specifically reserved for that order: any door that complies with the requirements can be used to fulfill that order. MTO doors for project mean that a door is reserved for a specific order the moment production starts. The MTS door types come in various configurations and have various specifications depending on the door type. There are approximately 82 configurations for the MTS doors, and for every one of these configurations safety inventory levels and maximum inventory levels have been defined. Before production starts, the operations needed for the production of the door are known and a rough estimate for the time needed for production of the complete door is generated, which is used for determining the moment of order-release. Each week approximately 8,000 doors are produced in the LDF, where the priority lies on the MTO doors and the remainder of the capacity is filled with orders for MTS doors. The capacity of the LDF is higher than those 8,000 doors per week, for example in week 26 of 2022 the production output was almost 11,000 doors. In Table 1 some parameters of the LDF are given. The data on which these numbers are based are from the period 1 November 2021 until 31 October 2022, so the span of one year. In this year, production stopped for four full weeks (Christmas one week and three weeks in the summer), and for a couple of national holidays. The production of the total number of doors, with a split between MTO and MTS doors, and the total deliveries are presented in Figure 1. More information about the differences between MTO and MTS and about batches is presented later in this thesis.

Doors are produced according to flow production, where the specific path depends on the complexity of the door configuration: some configurations need more operations or another variant of an operation than others. Some operations are carried out on a single machine, while others have multiple machines that can process only some configurations. In literature, this production paradigm is

	MTO	MTS	Total
Yearly number of doors	244,066	142,428	386,494
Average number of doors per week	5,085	2,967	8,052
Yearly number of Production Orders (POs)	60,632	6,917	67,548
Average number of production orders per week	1,263	144	1,407
Yearly number of batches	20,771	3,745	24,516
Average number of batches per week	433	78	511
Average number of doors per batch	11.8	38.0	
Average number of doors per PO	4.0	20.6	

Table 1: Production parameters LDF

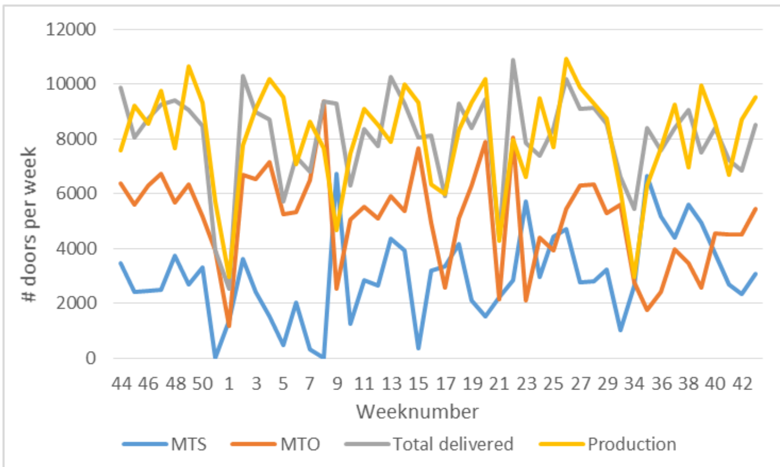


Figure 1: Weekly number of deliveries and production numbers

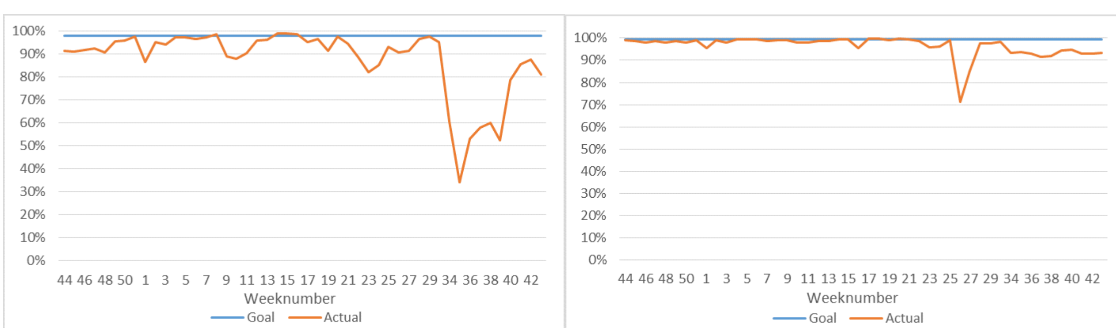
considered a hybrid flow shop. "A Hybrid Flow Shop (HFS) consists of series of production stages, each of which has several machines operating in parallel. Some stages may have only one machine, but at least one stage must have multiple machines." ([Linn and Zhang, 1999], p. 57). The operations that are undertaken in the factory are: sawing (optional, 3 parallel saws), labeling, pressing and filling (two separate machines), press braking, drilling, CNC machining and lock assembly (optional), painting, special assembly (optional), and packing. This process will be described in more detail later in this report. Production for most operations is done in two shifts per day, a morning and an evening shift, which result in a total of 16 hours per day available for production. Some operations work during the nights as well, and sometimes work is done on Saturdays if there is lack of capacity.

Berkvens works with the ERP-system BaaN, which indicates what doors need to be produced in what week. Based on these indications, production- and procurement recommendations are generated. These recommendations are evaluated and established, resulting in a first plan for production. The next step is to run the batching script, which is used to combine multiple production orders with similar specifications into one batch, with the goal of reducing handling time and save space in the factory, and in some cases reduce set-up times. When the batching script has run, the list with production orders and material requirements are established and handed over to the planning. There is no actual planning that states the order sequence and the order assignment to machines, but there is a list of orders and batches that need to be produced each day and the operators at each machine can determine for themselves which orders they produce. Most of the time there is either a very small set-up time between batches or a fixed set-up time regardless of the sequence of batches, and therefore the choice for the next batch does not heavily influence the production output of machines.

2.2 Problem description

In the previous section, the LDF has been described, which is considered a hybrid flow shop production environment. The LDF works with two due dates for the products, an internal and an external due date. The internal due date is set two days before the external due date, to ensure the external delivery reliability stays on a high level. The internal due date is only measured for MTO doors, while the external due date takes all doors into account that need to be delivered. The internal due date for producing lacquered doors is reached only 88.5% of the time in the period of 1 November 2021 until 31 October 2022, while the objective for this Key Performance Indicator (KPI) is set to 98% by Berkvens. This leads to a decrease in the external delivery reliability, as the external due date is reached only 96.4% of the time, while the objective for this KPI is set to 99.5% by Berkvens. In Figure 2 the weekly internal and external delivery reliability of the LDF are shown. The consequences of the low delivery reliability are disruption of other processes, both internal as well as external, and a decreased customer service level with possible penalties as a consequence. Internal disruptions are for instance an increased amount of rush orders that may increase lead times or lateness for other orders. External disruptions are for instance a disruption in the schedule of the mechanics that need to install the doors or delay on a building site due to late delivery of the doors.

Berkvens operates in the building industry, where disruptions in the original schedule often occur. The sales department calls customers about a week before the scheduled delivery date to ask whether the agreed upon delivery date should be maintained, or if the delivery date can be shifted backwards due to disruptions. Often, the customer agrees to shift the order backwards, which in theory could result in starting production of these doors later. However, due to the long makespan of the doors, the production of the door has already started before the shift in due date is processed, resulting in additional work-in-progress inventory or final inventory for Berkvens. Therefore, Berkvens would like to reduce the makespan of doors as well, to take advantage of shifting due dates. The problems Berkvens encounters are thus a lack of delivery reliability, both internal and external, and a makespan that is so lengthy that it prevents Berkvens from taking advantage of shifting due dates.



(a) Internal (b) External

Figure 2: Internal and external delivery reliability of the LDF

3 Research design

In this chapter, the research design is presented based on the problem described in the previous chapter. First, the research question is stated and sub research questions (SRQs) are presented, which aid in answering the research question. Then, the scope of the research is described. Next, the theoretical contribution of this thesis is explained, and lastly the methodology is given.

3.1 Research question

The two main problems encountered in the LDF of Berkvens are a low service level and a long makespan. The objective of this thesis is to develop a method that will improve the service level such that the actual service level will approach the KPI of 98%, with a secondary objective of improving the makespan. To reach these objectives, the main research question and several sub research questions are drawn up. The main research question for this thesis is as follows:

How to improve the service level and the makespan of doors produced in the lacquered door factory of Berkvens?

Sub research questions that help answer the main research question are the following:

1. What are the parameters and performances of the individual operations and how do the operations interact with and influence each other?
2. What distinction can be made between MTO and MTS orders regarding current performance and required performance, and how should the differences between those be addressed in the sequencing of orders?
3. How is the due date for production orders determined and what adjustments can be made in this based on lead times?
4. How is the production sequence of jobs and batches determined and what improvements can be made in the sequencing with the objective to minimize the tardiness and makespan of orders?

To answer the above questions it is important to define the scope of the research, such that it is clear for all stakeholders what elements are included in the research and what elements will not be considered. Therefore, the following section describes the scope of the research.

3.2 Scope

Defining the scope of the problem is important to limit the extent of the research area. Primary investigations on the potential causes mentioned in the cause-and-effect diagram in Appendix 1 have been carried out in the LDF. From this investigation it becomes clear that disruptions in the process due to failures or planned maintenance occur often and heavily influence both the makespan and the tardiness of orders. Furthermore, rework of products frequently occurs, approximately 6% of the doors experience some faults and either need to be repaired and be re-processed, or should be discarded and produced again from scrap. Besides that, multiple priority levels are distinguished: rush orders, new logistics concept (NLC), orders for France, and MTS orders. Rush orders are mostly orders that have been discarded and need to be produced again or orders of which the due date is coming up soon and for which it is still possible to be delivered on time. NLC orders have a lead time of 4 days and a hard due date because the salesperson calls the customers a week in advance to reconfirm the delivery date, which is then agreed upon again. The orders for France are prioritized next, because these are scheduled for transport and missing the due date would disrupt the transportation process, resulting in the need for more trucks and thus increased costs. MTS orders have the lowest priority because lateness of these orders results in the least disruptions for the other processes. Besides this, buffers exist between the operations and the capacity of those buffers is approximately 16 hours of production. This means that the actual buffer capacity differs between operations, but that for most operations a full day of production can be stored, which is almost always enough to prevent disruptions in the flow due to disruptions in one operation. However, when the sequencing of orders changes, the fill rate of buffers might change as well, and therefore the buffers should be included in this research. Lastly, the possibility of overwork is present, as some operations might also run during the nights or on Saturdays.

Shortages in the workforce were relevant until October 2022, leading to too little capacity to run all operations. However, from October 2022 onwards, the deficit in employees has been solved because employees from the HPL factory have been transferred to LDF due to diminished demand for HPL doors, which caused the HPL factory to switch from two shifts per day to only one. Even though the deficit in employees has been solved, overwork might still be required. Besides this, disruptions in the production due to lack of raw materials almost never occur, thus this is excluded from this research. Lastly, besides the different priority levels for the various types of orders, no distinction is made between customers, meaning that no customer has priority over another customer. Only very rarely an order for a particular customer is prioritized over other orders, resulting in a rush order.

Overall, failures and maintenance, rework, various priority levels, buffer sizes, and overwork are included in the scope of the research, because these factors may play a role with regard to the problem. The size of the workforce, raw material or production equipment shortages, and customer specific priority rules are disregarded, as they have no influence on the problem.

3.3 Literature review and theoretical contribution

A literature review has been conducted on the topic of hybrid flowshop scheduling [Jeuken, 2022]. From this review it became clear that many different methods can be applied to solve the HFS problem, which is the decision-making on the sequence of production and the assignment of jobs to machines in a HFS [Khalouli et al., 2008] [Li et al., 2012] [Sáenz-Alanís et al., 2016] [Ruiz and Vázquez-Rodríguez, 2010]. The HFS problem is NP-hard [Gupta, 1988], which means that solving the problem to optimality with an exact method could take too much computational time to be a realistic option. Therefore, the HFS problem can best be solved by a heuristic or a metaheuristic or a combination of the two, which is called a hybrid metaheuristic. In the literature review, several methods for solving the HFS problem were encountered, and it became clear that heuristics and metaheuristics were the most commonly used method for solving complex HFS problems. The most frequently mentioned heuristics were dispatching rules, which determine the order in which all jobs should be processed and have varying degrees of complexity [Hopp and Spearman, 2011a]. An example of a simple dispatching rule is the Earliest Due Date (EDD) rule, which prioritizes orders based on their due dates [Nahmias and Olsen, 2015]. More complex rules are often problem specific and take elements of the problem at hand into account when determining which order to prioritize. Multiple metaheuristics were mentioned, such as Simulated Annealing (SA), Tabu Search (TS), Genetic Algorithm (GA), Ant Colony Optimization, and Particle Swarm optimization, but the first three methods were most popular. SA is a local search heuristic that sometimes accepts inferior solutions to ensure the whole search space is included, ensuring that the best solution is accepted, and not just a local optimum is reached [Kirkpatrick et al., 1983]. TS has the same goal, but adds search spaces that are already investigated to a tabu list, which cannot be visited again [Glover and Laguna, 1998]. GA combines two solutions to a third one, the so-called child, and only the younger and stronger generations are included to search the whole search space [Portmann, 1996]. None of these methods had a clear advantage over any of the others. This is probably due to the fact that every method has its own advantages and disadvantages and the performance of the method depends a lot on the case at hand [Ruiz and Vázquez-Rodríguez, 2010].

The HFS problem has a standard form, which has the objective of minimizing the makespan and has the following parameters: machines and orders are available at time zero, machines at a given stage are identical, any machine can process one job at a time and one job can only be processed by one machine at a time, setup times are negligible, pre-emption is not allowed, unlimited buffer capacity between stages, and deterministic data of the system [Ruiz and Vázquez-Rodríguez, 2010]. The parameters of this standard HFS problem can be changed to have a better fit with a real-world case, which is often done in the reviewed literature. The changes in parameters that are encountered most often are the change to sequence dependent setup times, for instance in [Jungwattanakit et al., 2008], known job-to-machine allocations (see [Yalaoui et al., 2011]), and job re-entry, for instance in [Zhang and Chen, 2016]. Also, the objective can be changed to either a single-objective function that considers tardiness, as demonstrated by [Azadeh et al., 2019] and [Zhang and Chen, 2018], or a multi-objective function that considers both tardiness and makespan, as demonstrated by [Bozorgirad and Logendran, 2016] and [Lee et al., 2011].

In the literature review many real-world examples of the HFS problem with various assumptions and parameters were encountered. However, none of the papers found had the exact same manufacturing paradigm as Berkvens. The problem that is encountered is an HFS problem with the objective of minimizing the total tardiness and the makespan, in a large-scale system with finite buffer sizes, pre-emption of orders, and disruptions of the processes. For this specific HFS problem, an optimization project has never been carried out as far as the author is concerned. Therefore, researching this problem will contribute to both the theoretical field and the practical field.

3.4 Methodology

The methodology is based on a research cycle presented by [Bramoullé and Saint-Paul, 2010]: first literature research is done and hypotheses are presented, which is done in the first sections of this chapter. Then data is collected and analysed. Next improvement proposals are defined and tested. Lastly, the results are compared with findings in the literature and directions for further research are given. Below, the steps that are taken to follow the above research cycle and to answer the research question are briefly described.

Analysis of the parameters and performances of the current operations and their interdependencies: For sub research question (SRQ) 1, the parameters and performance of the current operations are mapped out first. Information that should be gathered here is for instance the production speed and capacity, number and cause of disruptions, number of parallel machines, and so on. Also, data about the past performance should be gathered and analysed to see how efficient the processes run. When this is done for all individual operations, the interdependencies between the operations should be investigated as well, to be able to investigate the performance of the whole flowshop. The results of this step are described in Chapter 4.

Investigation of the difference between MTO and MTS orders: For SRQ2 the differences between MTO and MTS orders need to be investigated. It is foremost important to know the differences between those, what role in the day to day operation of the LDF they play, and what influence these differences have. The results of this step are described in Chapter 4.

Investigation of the current sequencing procedure: For SRQ 3, the current procedure for sequencing the production orders and batches should be investigated. This should be done so the constraints and limitations of the flow shop will be clear and can be implemented in the to be developed methods as well, so these methods can practically be implemented in the company. The results of this step are described in Chapter 5.

Investigation of the current method to determine the due date and estimated lead time: For SRQ 3, the method for determining the due dates should be determined. Factors that should be considered are the adjustments that can be made and Berkvens is willing to make in this process. Furthermore, the estimated lead time is derived from the agreed upon due date, and investigations on how this estimation is established should be done to determine whether this estimation can be improved. The results of this step are described in Chapter 5.

Developing mathematical model for the encountered HFS problem: This step is included based on findings in the literature review. Here the conclusion was drawn that a mathematical model is a good starting point for solving any HFS problem, even though the model will probably fail to obtain good results in a reasonable computation time. The mathematical model will be tested on a small set of orders to determine its performance. The results obtained from the previous steps will be used as input for the model. The results of this step are described in Chapter 6.

Developing (meta)heuristic model(s) for the encountered HFS problem: Because solving a complex HFS problem with exact methods is likely impossible, heuristics and metaheuristics will be developed to solve the HFS problem. The goal is to develop several (meta)heuristics to be able to compare them. The heuristics that are encountered most often in the literature review [Jeuken, 2022] are variants of dispatching rules. The most-encountered metaheuristics in the reviewed literature are Simulated Annealing, Tabu Search, and Genetic Algorithm, and based on the findings on the processes, one of these models will be developed for Berkvens' case. The results of this step are described in Chapter 8.

Testing the developed models: The developed models should be tested in order to evaluate their performance. The performance of the models is highly dependent on results of the analysis done to determine the performance of the LDF. As the LDF is a complex network with various operations and waiting lines, a simulation model will be built to compare the performances of the various methods. The design of the simulation model and its validity and verification will be explained in Chapter 7. The results of this step are described in Chapter 9.

Drawing conclusions and answering research question In the last step, the main research question will be answered based on the findings of the sub-research question and the findings obtained in the other steps. The results of this step are described in Chapter 10.

4 Performance of the flow shop

In this chapter, the current performance of the LDF will be analysed. This is done to answer the first sub research question: "What are the parameters and performances of the individual operations and how do the operations interact with and influence each other?". This question is answered by gathering information and collecting data about the performance of the individual operations. This data is then analysed to investigate the performance of the individual operations. Lastly, the interdependencies between the operations are investigated.

4.1 Main operations of the LDF

In Figure 3 the flowchart of the LDF is shown. Here, the In this section the operations mentioned in this flowchart will be described in more detail and the main operations will be identified.

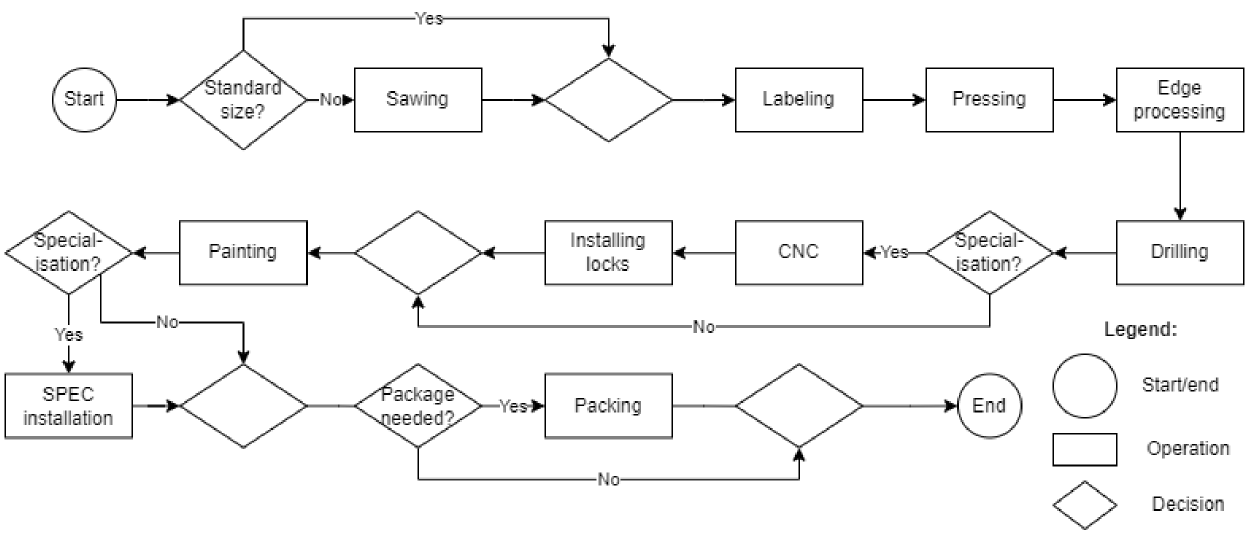


Figure 3: Production process of the Lacquered Door Factory

4.1.1 Sawing and Labelling

When a production order arrives, all the required preparation for the order is done. Thus all specifications, required materials, and other requirements are done. The firs step is to pick the materials needed from the inventory. A door consists out of four different types of raw materials: a filling, two plates, two lines, and two or three pillars. The raw materials are supplied by an outside supplier in standard measurements. Sometimes these measurements comply with the measurements needed for the door, while other times adjustments in the size should be made. When adjustments are needed, this is done by the sawing operation. There are various types of saws available: three saws for shortening which is used for both the lines (15% of the orders need this) and the pillars (7%), one saw for the plates (16%), and two saws for the fillings (12%). When all raw materials for a product have been gathered, one of the lines is labelled. This label is used to track the door in the factory, and to ensure the door follows the right route through the factory. During this stage of the process, one order is split in several sub-orders, which flow through the process simultaneously. Only when all parts required for one door are gathered and one of the lines is labelled, the product can continue to one of the presses.

4.1.2 Pressing

When all raw materials have the right measurements and are labelled, the pressing operation can begin. The pressing operation is the first operation where processing is done in batches, and from this process onward, the production is done in batches. There are two presses, the mass press (MP) and the small series press (SSP). The mass press is used for the doors that are standard, e.g. no windows or ventilation holes in the door or other complications. First, the materials are placed in the right position, after which the frame of the door and the bottom of the door is put together. Then, the door is filled and the top is attached, after which the door is pressed. The pressing is done so the glue that is used has enough time to harden. Then the door is put in the cooling line, where it cools down. After that, the doors are stacked on a plate, ready for transportation to the next station. From the start of the pressing process until the product is placed in the cooling line takes approximately 15 minutes. The cooling takes approximately 16.5 minutes.

The small series press works approximately the same: the individual pieces are put together with the bottom of the door, then filled, then the top is attached. The glued pieces are pressed together and then the cooling down starts. The main difference with the mass press is that more is done manually, the capacity of the press is smaller and the production rate is slower. Because of the manual actions, more specialized products can be produced on this machine, such as doors with a hole for a window or a ventilation hole. From starting the pressing process until the product is stacked takes approximately 20 minutes. The small series press does not have a cooling line, so the doors need to cool down while they are stacked, which takes approximately 12 hours before they can be processed further.

4.1.3 Edge processing

The edge processing (EP) is done in one big operation for all doors, where doors enter at one end, are processed, and come out at the other end. There are two types of doors: cover doors, which hang over the door frame, and stubby doors, which fall into the door frame. In the edge processing operation, the edges of the door are treated to ensure they have the right measurement, based on the type of door and the measurements of the door frame. Furthermore, the doors are milled and the edges are provided with edge tape. The edge processing operation is free of manual actions, apart from the supply and discharge of the stacks of doors. From the start of the edge processing until the product is stacked takes approximately 5 minutes.

4.1.4 Drilling

In October and November 2022 a new drill street has been installed in the LDF. This street simplifies the drilling process significantly: no more manual actions, and part of the locks can be installed already in the drill street. During the drilling operation, the holes for the hinges and the handles are made. Some doors need more holes than others, which depends on the type of door that is produced. The time needed for the drilling also largely decreased, as the new street is much faster than the old one, where many setups needed to be done manually. The new street is fully automatic, apart from the supply and discharge of the stacks of doors. From the start of the drilling process until the product is stacked takes approximately 3 minutes.

4.1.5 CNC and Locks installation

Some doors have special features, such as glass or ventilation grids or a specific pattern in the door. The operations that are required for ensuring these special features are mostly implemented by the CNC operation, which is a fully automatic process, apart from the supply and discharge of the stacks of doors. Only doors with special features need this operation. The CNC robot cannot operate when the door already contains a lock, so the doors that go through the CNC operation need to go to the lock installation operation afterwards. This is important, as the doors cannot enter the painting operation without a lock, which can be either a real lock or a dummy lock, because the doors will be damaged otherwise due to the pressure the doors encounter during painting. In the lock operation, locks are installed in the doors. The CNC installation works on only one door simultaneously, thus the time it takes for one product to flow through depends on the production rate, which will be determined later on in this chapter.

4.1.6 Painting

During the painting operation, the doors are sanded and painted on both sides. This is done in the paint street, which is fully automatic except for the supply and discharge of stacks of doors. Each side of the door is sanded, painted, varnished, and blown dry. This is done in multiple stations that are connected by a conveyor belt. Furthermore, the quality of the paint is also checked, and a door is repainted if it fails to meet the quality standards. A door is painted on both sides, so it needs to go through the street twice, which is done automatically as well. Multiple colors for doors are offered by Berkvens, and whenever a color change is needed, the line needs to be empty and another color should be installed, which takes approximately 30 minutes in total, 12 for emptying the line and 20 for the changeover. Furthermore, whenever the employees take a break, the line should be empty as well, which also takes 12 minutes. The time it takes for one product to flow through the painting operation is approximately 12 minutes as well.

4.1.7 SPEC installation

The special features are installed next for the doors that need it. This is done in the SPEC installation operation. This is all manually done with the help of simple tools. There are several operations that can be done here, such as setting glass in the doors or installing the ventilators. The SPEC department is physically small, there is often lack of space in the buffer between the painter and the SPEC installation department. This results in quite some time spent for searching the right doors or moving around doors. Furthermore, the SPEC department is one of the departments that actually needs specialist operators, which are hard to come by. The demand for the SPEC department often varies over the weeks, and therefore SPEC operators are often exchanged with the HPL factory. On average, there are two operators per shift working in the SPEC department, and the department runs on two shifts per day. In the SPEC department, one operator works on one product simultaneously, thus the time it takes for one product to complete the SPEC operation depends on the production rate.

In 2023 the SPEC operation will undergo some significant changes. The goal is to physically move the SPEC department to the HPL factory, where there is a higher degree of automation and more space to carry out SPEC work. Currently, this change is gradually carried out, with some operations moving to the HPL factory already. The goal is to realize this changeover completely after the summer break of 2023.

4.1.8 Packing

After the SPEC department, or the painting department if the SPEC installation is skipped, the door is ready for packing. Most doors need a package, but some doors may skip this operation. In the packing operation, the doors are sealed in plastic and stacked on a pallet. The pallets may be stacked as well based on transport orders. When all operations are done, the door is reported ready and transported to the final goods warehouse. The time it takes for one product to flow through the packing operation is approximately 5 minutes.

4.2 Data gathering

Many data sources and -bases are present within Berkvens, that are not all interconnected. This means that data is gathered from multiple sources, and a link between those sources should be established to evaluate the reliability of the data. Five different sources have been used to gather data, which are shortly described below. Production data has been gathered over the period from 1 November 2021 until 31 October 2022, such that there is data about the most recent full year.

ERP system: Berkvens uses an ERP-system called BaaN, which is used company-wide. Berkvens has a need to change to a new ERP system, as they feel the current system is outdated. This is the main reason for the switch to a new ERP-system by the start of 2024. The data in BaaN is not 100% accurate according to the people that often work with BaaN. The orders (both production and sales orders) have a lot of detail, for instance the route, the delivery date, the batch number in which the order was produced, the size of the order, and the theoretical lead time can be found, along with many specification for the type of door being produced. However, this overview is not fully complete nor fully accurate, which complicates this research significantly. Two main overviews have been gathered via BaaN: an overview of the production orders with an external delivery date between 01-11-2021 and 31-10-2022, and an overview of the production orders that are reported as finished in this period. The first overview is much more elaborate, but does not include the finished production date, while the second overview complies more with the data gathered from other sources and the estimated numbers by the operations manager. Besides these two main overviews, additional data about the products has been gathered, such that the product can be classified as an MTO or an MTS door.

Maintenance Control: Maintenance Control is an application that the technical service department of Berkvens uses. In this application, the maintenance orders are registered and planned. An overview of the maintenance activities of the machines of the LDF has been gathered, with the goal of determining the standstills per machine in the investigated period. Furthermore, the planned maintenance can be gathered from this overview as well, which can be useful for determining the available production time in the future.

Performance Analyser: Performance Analyser is an application in which the performance of several operations is registered. For every shift the run time and the disruptions are registered: if an operation has no output for a certain amount of time, a disruption is automatically added to the log. The operators should denote the cause of the disruption, although they sometimes fail to do this, thus performing a reliable analysis on causes of disruptions might be difficult.

Production output individual operations: Every day, the team leaders of the factory floor put the numbers about the production output of the previous day in an Excel sheet. This is done manually and based on the actual output. In this file, for most of the operations, also the number of doors that have quality issues are mentioned for most operations.

Delivery reliability data: The internal delivery reliability is calculated in an Excel file, in which the number of ordered doors and the actual delivered doors per week are recorded. In this file, also the production output is mentioned per week.

Data from these five sources is gathered, compared, and combined to determine what data source seem reliable and what data can be used for further analyses. The data from the delivery reliability file and the production output of the individual operations, foremost of the packing and total orders finished, comply fairly well with each other. In an ideal world, the data gathered about the completion day that is registered in the ERP-system should be approximately equal as well. The data from these four sources has been plotted in Figure 4, from where it can be seen that the data from the delivery reliability file and the total completions data based on the data from the individual operations file are completely equal. The data gathered from the packing machine follows quite closely as well, which is logical because not all doors are packed. The data from the ERP-system follows approximately the same pattern although clearly there are differences between the data from the different sources. Because the data from the delivery reliability file and the individual operation file are the same, this data is regarded reliable. However, this data does not have any specific information about the order type, route, delivery date, and specific details. Therefore, the data from the ERP-system is used as well for gathering more information about the specific orders, such as type of order, route, estimated lead time, etc.

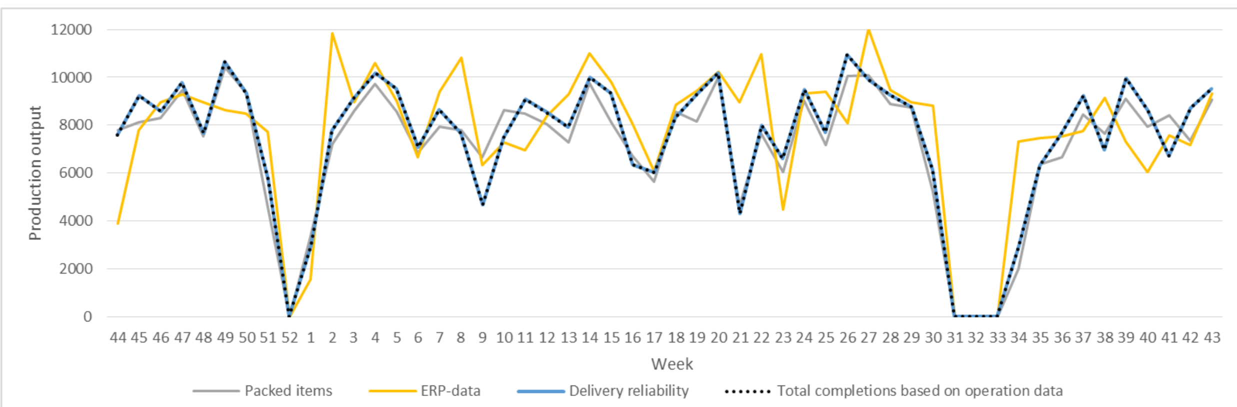


Figure 4: Comparison production output registered in various databases

4.3 Performance of the main operations

The main operations are sawing, labelling, pressing, edge processing, drilling, CNC, painting, SPEC installation, and packing. The other operations are important for the flow, but are neither mechanized nor a bottleneck. For all operations except sawing and SPEC installation, some information regarding daily production such as production time, standstills, maintenance, faulty doors, and production output are recorded, because those operations are mainly automatic. Sawing and SPEC installation require more manual actions, and therefore less information is recorded about those operations, which explains the need for assumptions and estimations for these processes. Furthermore, the new drill street changes the parameters of the drilling process significantly, and therefore also assumptions and estimations for this process need to be made. Below, more information on all operations regarding performance is given.

4.3.1 Performance of presses, edge processing, CNC, painting, and packing

As mentioned in the scope, failures, maintenance, rework, various priority levels, and overwork should be taken into account when answering the research question. When looking at the performance of the individual operations, mainly failures and maintenance should be taken into account when determining the production rate, as failures and maintenance are reasons for standstills and hence a smaller production capacity. Thus data has been gathered about the production output per operation in the period between 01-11-2021 and 31-10-2022. This data is fairly accurate, as described in the previous section. However, this data fails to specify the type of order, MTS or MTO, as well as the priority level of the order, which makes it impossible to determine whether different production rates are used for different types of orders or priorities. It is therefore assumed that all products are produced with similar production rates per operation.

Data is also available on the standstills in the process, which are mainly due to maintenance, both planned and unplanned. This data comes from a system called maintenance control, where all disruptions in the process that needed a service engineer from the maintenance department are recorded. More data about disruptions in the process that are not due to maintenance is available in a system called Performance Analyser. This system automatically records disruptions when the production of doors is interrupted for more than two minutes. The disruptions are categorized in broad categories, such as logistics, employees, start-up/shut-down, process technical, but also non-defined for the disruptions where the operators forgot to fill in a cause. The data from Maintenance Control and Performance Analyser are compared and the amount of disruptions based on planned maintenance and their duration are determined.

The daily available time for production for the main operations has been determined, which is based on the number of shifts. Most operations work with two shifts of 8 hours each excluding breaks. This means that for most operations, there is a daily capacity of 16 hours. Some operations work with three shifts, such as the CNC machine, which can run without operators during the night when enough doors are available for production. Some operations might work on Saturdays when demand is high, but only simple doors are done during these hours, so those are not representative for normal production days. Therefore, production rates are only determined based on data from Monday to Friday. The LDF was closed from 23-12-2021 until 5-1-2022 to celebrate Christmas and New Year and during the Dutch Building vacation in the summer, which lasted from 29-7-2022 until 21-8-2022, so those days are also removed from the data. Furthermore, production stops occured on 28-2-2022, 18-4-2022, 27-4-2022, 26-5-2022, 27-5-2022, 6-6-2022, and 16-6-2022, so those days are removed from the data as well. Besides those days, also so-called "Conjunctuur" days exist, which are a couple of days during the year when the managers can determine that production is stopped because there is low demand. This happened on 4-2-2022, 11-2-2022, 18-2-2022, and 25-2-2022 in the period from which data is gathered. The production did not fully stop on these days, only one shift had to work, so the available production time was halved.

The daily production per operation and the daily available time of the operation have been compared for the operations where this was possible. By dividing the available time by the number of products produced for each day, the production rate per hour was determined. Boxplots of the rates are shown in Figure 5. From these boxplots it can be clearly seen that the small series press and the CNC machine have many outliers with a high hourly production rate compared to the other datapoints. For the CNC this can be explained by the nature of the operation: some doors need a lot more time on the CNC based on their design, because they have glass plates, lines, ventilation grids, etc. The more things need to be done during the CNC operation, the more time is needed for the operation and the smaller the hourly production rate is. As batches are formed based on routes and

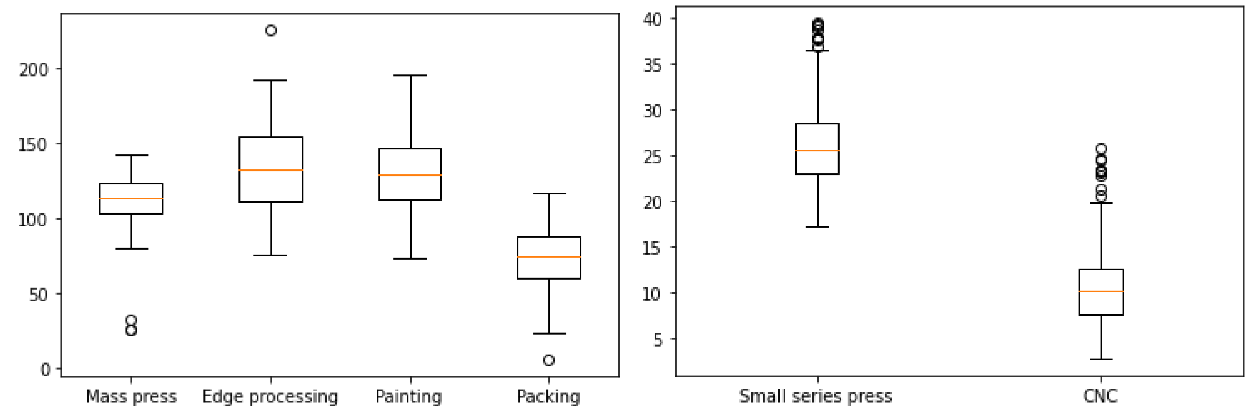


Figure 5: Production rates per hour

specifications, it is logical that doors that need lines are done in badges, which requires significantly less time than doors that need a hole for a ventilation grid, which is where the outliers come from. The many outliers for the small series press can be explained similarly. Different configurations lead to varying production times when assembling the door frames. Therefore, some batches require a lot of time in the press process, while others require less time. The other outliers encountered according to the boxplot, such as the low values in mass press and packing, are events where no explanation can be found for specifically. There might be a reasonable explanation for those outliers, but no cause or disruption has been registered, and hence all outliers should be included in the data set.

To be able to estimate production rates for future batches of doors, the underlying distribution that the production rates follow should be determined. The first step to determine this is to fit various distributions to the data and see what distribution has a good fit and what the parameters of the fitted distribution are. To fit distributions to the data, the data is imported in Python and the Fitter library is used. The Fitter package identifies the underlying distribution from which a data sample is taken by comparing multiple distributions, and it estimates the parameters of the potential distributions [Cokelaer, 2019]. The results obtained by the Fitter for the individual operations are displayed in Figures 7 until 11. In these figures, the x-axis displays the value and the y-axis the proportions with which these values are encountered in the data.

Based on the production rates, the production time per product can also be determined for each operation: by dividing one with the hourly production rate, the time needed per product in hours is obtained. This data might have a better fit with a distribution than the production rates, and therefore the production time is also checked with the Fitter. In Figure 35 until 40 appendix 2 the results obtained by the Fitter can be found.

The next step to determine the underlying distribution of the hourly production rates of the operations is to perform a goodness-of-fit test for the fitted distributions. For this, the Kolmogorov-Smirnov test for goodness-of-fit is used, as this test is non-parametric and is widely used and proven [Massey, 2012]. It is logical that all operations follow the same probability distribution. From the figures obtained by checking the distribution fit it becomes clear that the Normal distribution has a good fit with most of the operations. Therefore, the goodness-of-fit test is done with the normal distribution. Here, the null hypothesis (h_0) is as follows: the hourly production rate follows the Normal Distribution. The goodness-of-fit test is performed in Python as well, by using the

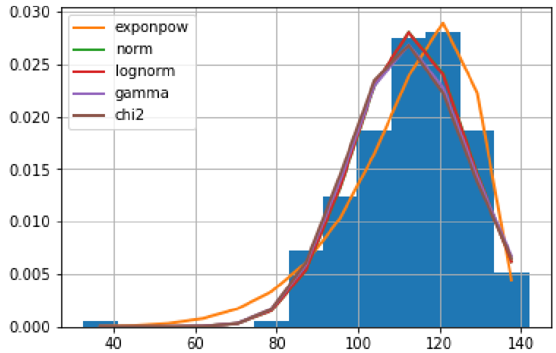


Figure 6: Hourly production rate mass press

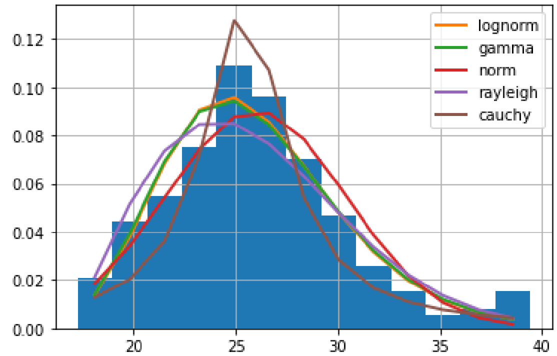


Figure 7: Hourly production rate small series press

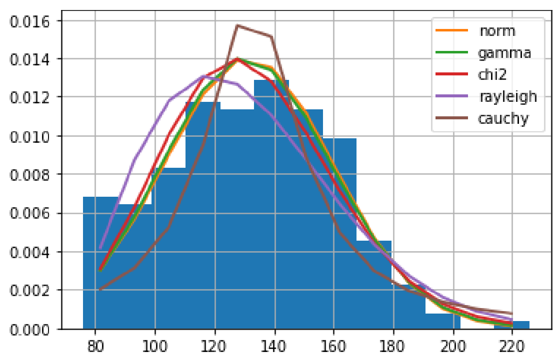


Figure 8: Hourly production rate edge processing

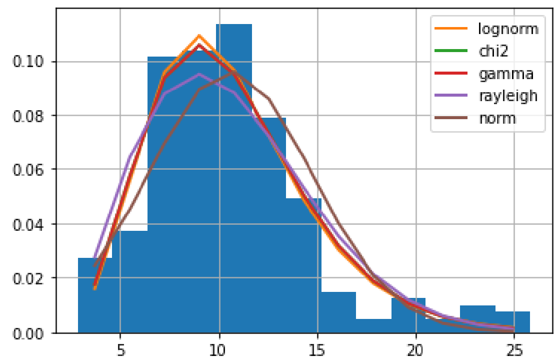


Figure 9: Hourly production rate CNC

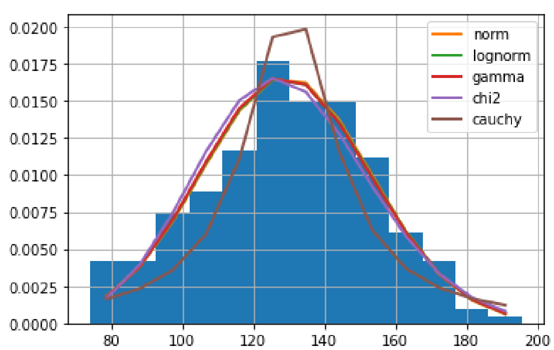


Figure 10: Hourly production rate Painting

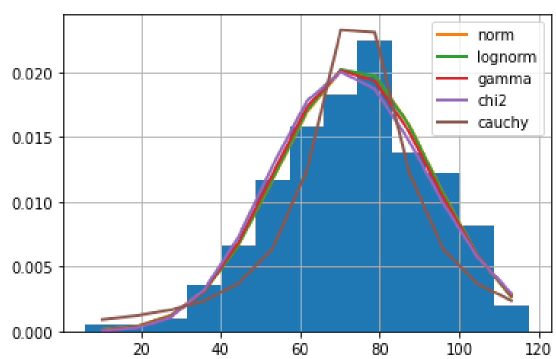


Figure 11: Hourly production rate Packing

scipy.stats.kstest function. This function compares the underlying distribution of two independent samples [Massey, 2012], which means that the data about the production rate is compared against sampled data from a Normal distribution that takes the estimated parameters from the actual data. The goodness-of-fit test is done with a confidence level of 0.95, i.e. $\alpha = 0.05$, so if the P-value is larger than 0.05, the null hypothesis cannot be rejected and the data may well follow the Normal distribution with the estimated parameters.

The results of the goodness-of-fit test for the hourly production rates are reported in Table 2. From this table it becomes clear that the null hypothesis is accepted for all operations, which means that it is safe to assume that the hourly production rates of these operations follow the Normal distribution with the estimated parameters mentioned in the table. A goodness of fit test with the normal distribution has also been done for the production time per product, and the results of this test are presented in Table 32 in Appendix 2 as well. From these results it can be concluded that only the presses follow the normal distribution with the given parameters. However, based on the results obtained by the fitter, the production time per product for the operations seem to have a good fit with the Gamma-distribution. Therefore a goodness-of-fit test on the production time per product has been done as well with the Gamma distribution. The results are reported in Table 3. Normally, the Gamma distribution has two parameters, namely a shape parameter and a scale parameter. The Fitter library recognizes a third parameter, namely the location (loc) parameter, which is the point from which the distribution starts, which is normally 0. From Table 3 it becomes clear that the null hypothesis is accepted for every operation as well, which means that the data could well follow the Gamma-distribution.

The output of the goodness-of-fit tests with the Normal and the Gamma distribution is compared. It would be logical that the main operations all have the same distributions with regard to the production rate/time, as products are processed similarly. Therefore, a choice should be made to either have hourly production rates with the Normal distribution or production times per product with the Gamma distribution. Because products in different batches are independent of each other, the production rate for one batch will likely differ from the production rate of another batch produced in the same hour. Therefore, production time per product seems a more logical way to express production time. Thus the production time per product will be used to determine the performance of the LDF later on.

Operation	**Mean (μ)**	**StDev (σ)**	**P-value (norm dist.)**	h_0
Mass press	113.05	14.21	0.5436	Accepted
Small series press	25.99	4.42	0.0709	Accepted
Edge processing	131.56	28.45	0.6732	Accepted
CNC	9.86	3.02	0.7389	Accepted
Painting	129.25	23.94	0.9102	Accepted
Packing	73.66	19.55	0.9693	Accepted

Table 2: Results goodness-of-fit test with Normal distribution

Operation	**Shape**	**Scale**	**Loc**	**P-value (gamma dist)**	h_0
Mass press	3.62	0.0006	0.0068	0.9823	Accepted
Small series press	91.62	0.0007	-0.0251	0.6786	Accepted
Edge processing	4.30	0.0009	0.0041	0.5799	Accepted
CNC	2.96	0.0264	0.0319	0.1414	Accepted
Painting	3.85	0.0008	0.0048	0.6442	Accepted
Packing	2.05	0.0030	0.0084	0.9732	Accepted

Table 3: Results goodness-of-fit test Gamma distribution production times per product

4.3.2 Performance of sawing and labelling operations

For the sawing operation, no data is saved in any system, so the production rates cannot be determined in the same way as done for the main operations in the previous section. It is however good to have an estimate for the production time needed for this operation, to be able to represent the flow shop as good as possible. In April 2021 a student from Fontys' Industrial Engineering program conducted a study in which he conducted time measurements of sawing operations. Since then, nothing has changed in the sawing process, so these time measurements can still be used today. For each of the four sawing operations, sawing fillings, lines, plates, and pillars, 30 products were followed and for each product the time for the sawing operations and the labelling was measured. These time measurements include additional handling of the goods, so it actually represents the total time needed to complete the operation. The time needed per product is calculated by dividing the time needed for the batch by the number of products in the batch. Then the hourly production rate is calculated from this, and the resulting data is used to fit distributions and conduct a goodness-of-fit test in the same way as explained in the previous section. The boxplot and the results of the fitter are shown in Figures 12 until 17.

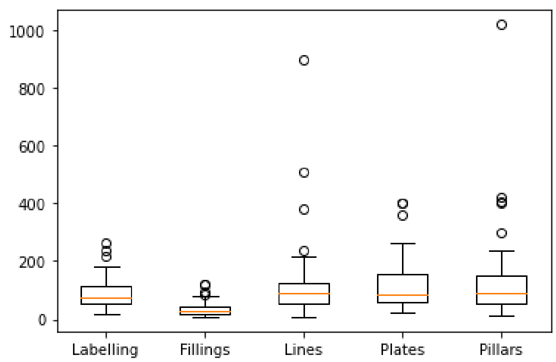


Figure 12: Production rates per hour sawing and labelling

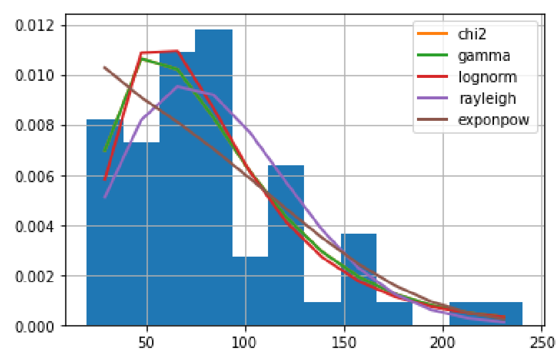


Figure 13: Hourly production rate labelling

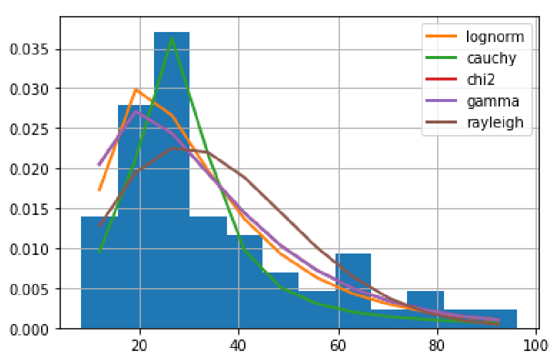


Figure 14: Hourly production rate sawing fillings

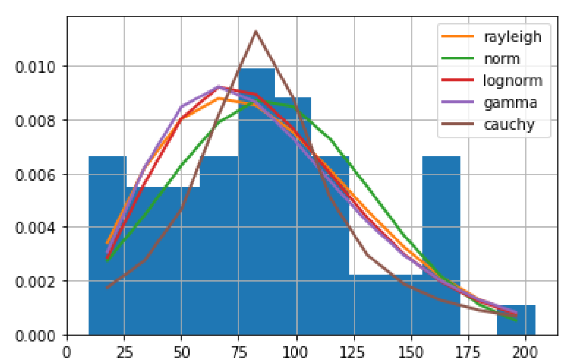


Figure 15: Hourly production sawing lines

When looking at the boxplot, it becomes clear that there are quite some outliers, especially in sawing lines and sawing pillars. However, there is no additional information on why these outliers occurred and hence these data points should be included in the data set. When looking at the results of the fitter, it becomes clear that the Normal distribution fits the data badly for all five operations.

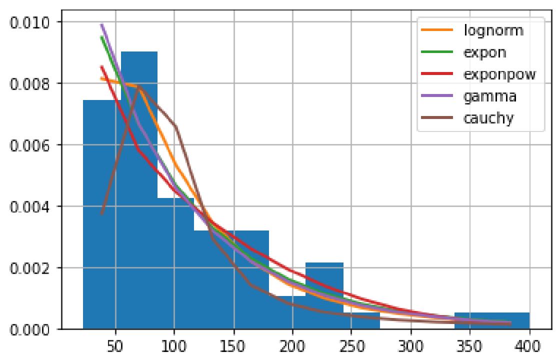


Figure 16: Hourly production rate sawing plates

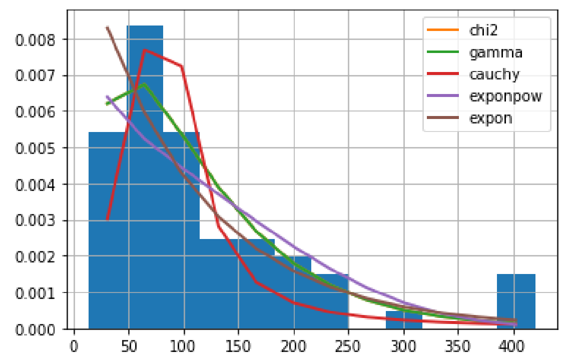


Figure 17: Hourly production rate sawing pillars

However, the Exponential distribution seems to have a good fit with most of the operations, thus the Kolmogorov-Smirnov goodness-of-fit test has been done with the Exponential distribution. Here, the null hypothesis (h_0) was: the hourly production rate follows the Exponential distribution. The results of the goodness-of-fit test are shown in Table 4, where the location parameter again indicates the start of the distribution. Based on the obtained results it can be concluded that the null hypothesis is accepted for all operations, and it is thus safe to assume that the hourly production rates follow the shifted Exponential distribution.

	Loc	**Scale (λ)**	**P-value**	h_0
Sawing fillings	8.35	28.77	0.1615	Accepted
Sawing lines	9.68	107.15	0.2944	Accepted
Sawing plates	23.08	93.14	0.5849	Accepted
Sawing pillars	13.71	117.02	0.2257	Accepted
Labelling	20.00	67.32	0.0536	Accepted

Table 4: Results goodness-of-fit sawing and labelling with Exponential distribution

4.3.3 Performance of drilling operation

In Section 4.1.4 the drilling operation has been described. Drilling is a main operation: almost all doors flow through the drill street. As stated in the description, a new drill street has been implemented in October and November 2022. In December 2022 this drill street was fully operational, meaning that the old drill street is stopped. The implementation of this new drill street results in inaccurate data about the drilling operation, because the data that has been gathered about the drilling operation is obsolete. The number of doors that require a drilling operation remains the same, but the time needed to perform the operation changes. No accurate data is available yet, because the new drill street encounters teething problems, and hence the data it produces is unreliable.

To be able to analyse Berkvens' flow shop, assumptions are made about the performance of the drilling operation. The cycle times for specific types of doors have been gathered, and based on the number of doors of these specific types that were produced in the period of which data was gathered (1-11-2021 until 31-10-2022), estimates were made about the time needed for these doors had they gone through the new drill street. The cycle times were increased with a factor of 15% per product to account for disruptions in the process. This number was chosen in collaboration with company experts that were involved in the implementation and operation of the new drill street.

Cycle times can be either 80 seconds, 25 seconds or 20 seconds per product, depending on the type of product. Increasing the cycle time with 15%, this becomes 92 seconds, 28.75 seconds and 23 seconds respectively. The proportion of the products that have these cycle times are 3.2%, 16.4%, and 80.4% respectively. This results in an average cycle time of 26.17 seconds with a standard deviation of 2.11 seconds per product. This translates to an hourly production rate of 137.56 products. These numbers will be used when estimating the throughput of the drill street in the future.

4.3.4 Performance of SPEC installation

As described in Section 4.1.7, various types of installations are done in the SPEC department, such as installation of ventilation or glass windows in the door. No data is available on how long the installation of various specifications takes, as this operation is mainly manual and nothing about duration or output is registered. In the past, the SPEC department has worked with norm times, which are used here to estimate the throughput per hour. Data has been gathered about the number of products that needed a certain type of specialisation by looking at the specifications of the products. However, this data contained errors and missing values, and lacked some information. Therefore the numbers of product that needed a certain operation is expressed in percentages rather than actual number of doors, and these percentages are based on findings in the data, manually corrected in cooperation with the foreman of the SPEC installation, the head of the Exhibitions Planning Office and the Operations Manager. The percentages and the throughput in doors per hour are given in Table 5. Based on these numbers, the mean (μ) and standard deviation (σ) are calculated with the following formulas: $\mu = \sum x \cdot P(x)$, and $\sigma = \sqrt{\sum (x-\mu)^2 \cdot P(x)}$. This results in a mean of 9.62 products per hour with a standard deviation of 5.84. Note that there are two operators working per shift, and therefore the throughput is 19.24 doors per hour.

SPEC type	Doors per hour (x)	Percentage ($P(x)$)
Ventilation	12.5	28.5%
Glass	2.5	33.3%
T-lat	7	6.9%
Drop sill	8.5	7.8%
Helios	10.5	0.4%
Yacht varnish	17.5	23.0%

Table 5: Norm rates SPEC

4.4 Standstills of machines

As mentioned in Section 4.3.1, data is available on the standstills of the process for the presses, edge processing, CNC, painting, and packing operation. Information about these standstills is gathered, namely how often a standstill happens and what the duration of the standstill was. This is done in order to determine the available time for production, which is a limitation of the production process and should be taken into account.

For the number of times an operation has a standstill, data is gathered about the standstills per day. The number of days in the period 1-11-2021 until 31-10-2022 for which a standstill was registered has been counted, and the percentage of days with a standstill is calculated. Furthermore, the length of the standstill in minutes is registered and a Kolmogorov-Smirnov test has been performed for the duration of the standstill. For this test, the data is fitted with the exponential distribution, because this distribution seemed to have a good fit with most of the operations. The null hypothesis is thus: the duration of standstills follows the Exponential distribution. The results of this analysis are stated in Table 6. From this table it can be observed that the exponential distribution seems a good fit for all machines as the null hypothesis is accepted for all machines.

Operation	**Number of days with standstill**	**Percentage days with standstill**	**GoF Expon. Dist.**			
			Location	**Scale**	**P-Value**	h_0
Mass press	12	5.33%	240	123.8	0.6482	Accepted
Small series press	5	2.22%	60	480.0	0.6287	Accepted
Edge processing	76	33.78%	1	143.8	0.0582	Accepted
CNC	16	7.11%	15	56.3	0.4941	Accepted
Painter	59	26.22%	10	244.9	0.3161	Accepted
Packer	45	20.00%	15	72.3	0.429	Accepted

Table 6: Results analysis machine standstills

4.5 Order- and product arrivals

The amount of orders and products that need to be produced varies over the weeks, and the average makespan in a busy week is probably higher than that of a quiet week. Furthermore, the chance of late deliveries is also higher in a busy week, because the capacity usage of the factory is higher and the whole facility is thus prone to larger deviations. Therefore the distribution of the arrivals of production orders and individual products is investigated in this section. This has been done for the total orders and products, but a distinction is also made between MTO and MTS orders and products, to see if there are differences between those.

In section 4.2 it has been concluded that the data from the ERP-system is not the most reliable source. However, to be able to determine the arrivals of orders this data should be used, as this is the only data that has information about the type of orders, the theoretical leadtime, and the external due date. The arrival of the production orders is determined by subtracting the theoretical leadtime from the external due date, which results in the order-release-day, which is the arrival day of the production order. The number of MTS- and MTO orders and products and total orders and products arrivals per day has been determined for the period between 01-11-2021 and 31-10-2022. After some investigation, it seemed that the end of this period has missing values, which is logical: the data is extracted from the ERP-system based on orders that have been finished before 01-11-2022, but orders that arrived late in October are not yet finished at that moment, thus there are missing values in this period. After some investigation, it seemed that most of the orders that arrived on or before 14-10-2022 were finished by 31-01-2022, so data from the period 01-11-2021 till 14-10-2022 is used for the analysis.

A boxplot is made for the MTS, MTO, and total orders, which is shown in Figure 18. The same has been done for the individual products, see Figure 19. When looking at these boxplots, it becomes clear that there are far fewer MTS orders than MTO orders, but that the number of MTS products is only slightly less than the MTO products. This implies that there are more products in an MTS order than in a MTO order. Besides this, it is clear the the number of outliers is higher for MTS orders and products than for MTO, which might be explained by the fact that more MTS products are made when the demand for MTO products is low, which means the demand for MTS products has higher fluctuations. All demand for MTS orders is internal, generated by the ERP-system when the inventory level drops beneath a certain number of products.

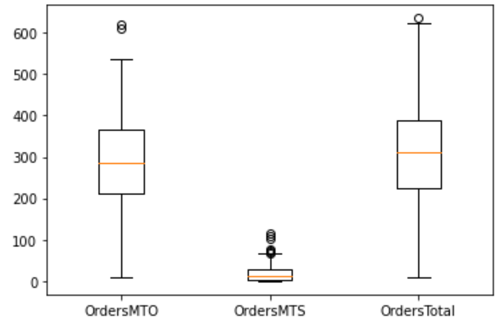


Figure 18: Daily order arrivals

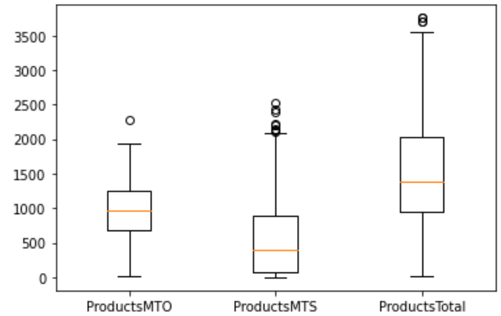
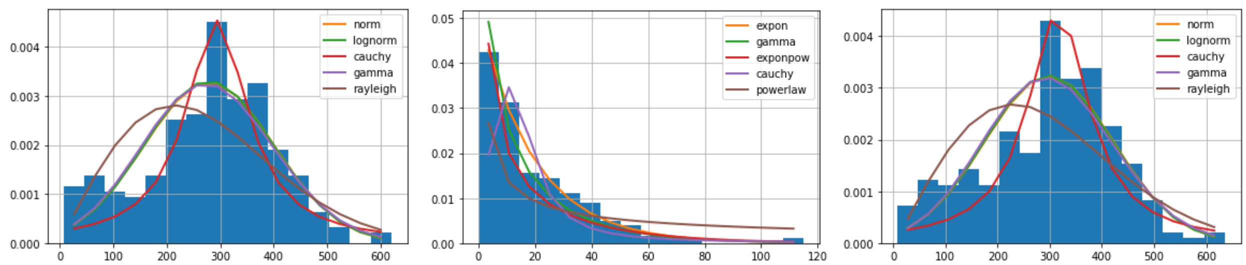


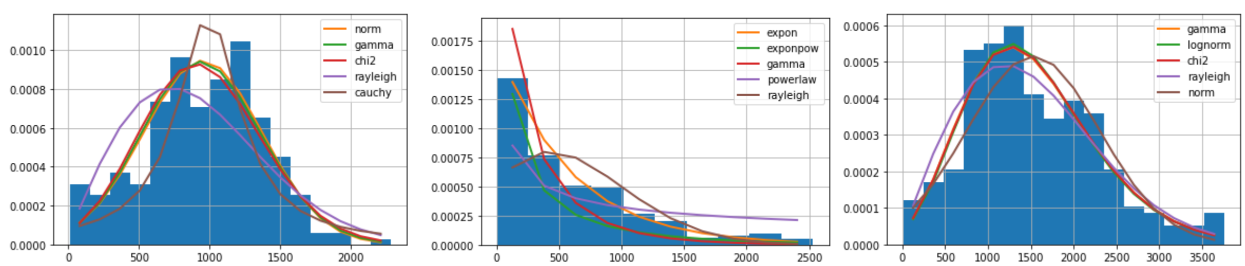
Figure 19: Daily product arrivals

The Kolmogorov-Smirnov test for goodness-of-fit has been done for the daily MTO, MTS, and total order arrivals, and for the daily MTO, MTS, and total product arrivals. First a fitter is used on the six types, and then the goodness-of-fit test has been done for both the Normal distribution and the Gamma distribution. The results of the fitter for the order arrivals is shown in Figure 20 and for the product arrivals in Figure 21. The results of the goodness-of-fit test are shown in Table 7.



(a) MTO order arrivals (b) MTS order arrivals (c) Total order arrivals

Figure 20: Daily order arrivals



(a) MTO product arrivals (b) MTS product arrivals (c) Total product arrivals

Figure 21: Daily product arrivals

When looking at the results of the goodness-of-fit tests, see Table 7, it seems that both the Normal and the Gamma distribution have a good fit or the MTO and Total orders and products. However, only the Gamma distribution fits with the MTS order arrivals, and neither fits the MTS product arrivals. In fact, no distribution fits the MTS product arrival data.

	Norm dist			Gamma dist			
	Mean	**StDev**	**P-Value**	**shape**	**scale**	**loc**	**P-Value**
Orders MTO	279.3	121.4	0.2001	388.3	6.2	-2143.5	0.1208
Orders MTS	19.4	19.5	0.0000	0.7	25.1	0.0	0.0000
Orders Total	298.7	123.5	0.1085	364.6	6.5	-2084.4	0.0563
Products MTO	951.3	421.3	0.5746	554.3	18.0	-9031.2	0.4396
Products MTS	576.1	574.1	0.0000	0.6	476.2	0.0	0.0000
Products Total	1527.4	774.1	0.0802	8.3	270.0	-711.9	0.9170

Table 7: Results goodness-of-fit tests arrival distributions

4.6 Relations between the operations

In this section, the relations between the operations are analysed. First of all, the rework and repair of products is investigated and analysed. Next, the first part of the flow in the LDF is analysed, which is the order release until the moment the order is ready for the press. Then the second part of the flow in the LDF is analysed, which is from the moment the order is ready for the press until the moment is ready for transportation. A split is made between these two, as in the first part of the operation a product can be processed by multiple operations simultaneously, while in the second part of the operation the products flow in one piece, as the door is assembled there. Furthermore, the products flow in batches in the second part of the operation, which alters the characteristics of the LDF significantly. When both parts of the flow are analysed, the main routes products take are investigated. Lastly the arrival rates per operation are determined, to check whether the operations are stable.

4.6.1 Rework and repair

Unfortunately, rework and repair of products is unavoidable within the LDF; doors get damaged and should be repaired or discarded and produced from scrap. This influences the process, as the flow of products might be reversed or products need to flow through multiple operations twice. Furthermore, the process gets disrupted in other ways as well, because the due date of products stays the same, so products might get a higher priority level to ensure the due date is reached. This also influences the delivery reliability and the makespan, because the products spent longer in the system, hence the increase in makespan, but this might lead to late deliveries as well. It is therefore important to take rework and repair into account when determining the relations between the operations.

Data about rework and repair has been gathered over the period from 01-11-2022 until 31-10-2022. The percentage of products that need to be either repaired or discarded is calculated from this data, and the results are presented in Table 8. Here, the percentage of products that need repair or discard are both specified per operation and overall. As not all products flow through every operation, the sum of percentages of individual operations differs from the overall percentages. From this table it can be observed that the number of repairs is higher than the number of discarded products, which is fortunate, as discarded products disturb the flow more than repairs. Furthermore, foremost the painting operation and the SPEC operation have a high number or products that need some form of rework. The total percentage of products that need some form of rework is 6.4%.

Operation	Repair	Discarded	Total
Sawing + Labelling	0.0%	0.0%	0.0%
Press	0.0%	0.0%	0.0%
Edge Processing	0.4%	0.1%	0.6%
Drill	0.9%	0.4%	1.2%
Locks + CNC	0.2%	0.4%	0.5%
Painter	2.6%	0.9%	3.5%
SPEC	2.8%	1.8%	4.6%
Packer	0.4%	0.3%	0.7%
Overall	4.4%	2.1%	6.4%

Table 8: Rework and Repair percentages

4.6.2 Order release until order ready for press

Before, it has been stated that the process in the LDF is a hybrid flow shop. Recall from Section 3.3 that in the standard HFS problem one job can only be processed by one machine at a time, which means that the order cannot be at multiple operations at one time. For the most part this is true, but for the beginning of the process, the sawing and labelling operation, the order can be handled by multiple machines at the same time, as the door is not yet assembled there. In Figure 22 this split is visualized by a Business Process Model. Here, the AND-nodes represent a split of the order into the four sawing operations, which can be performed but not necessarily need to be done. The percentages after the XOR-nodes indicate what percentage of the orders actually follow the path, e.g. 15.3% of the orders need the Sawing Lines process, while the other 84.7% of the orders can continue to the labelling process immediately. The labelling is always done when the sawing of lines is done, if this is done, and otherwise after the lines are gathered. This is because the label is placed on one of the lines. When all four sub-processes come together again, this means that all the materials are available and the order is ready for the press, where the process becomes a HFS.

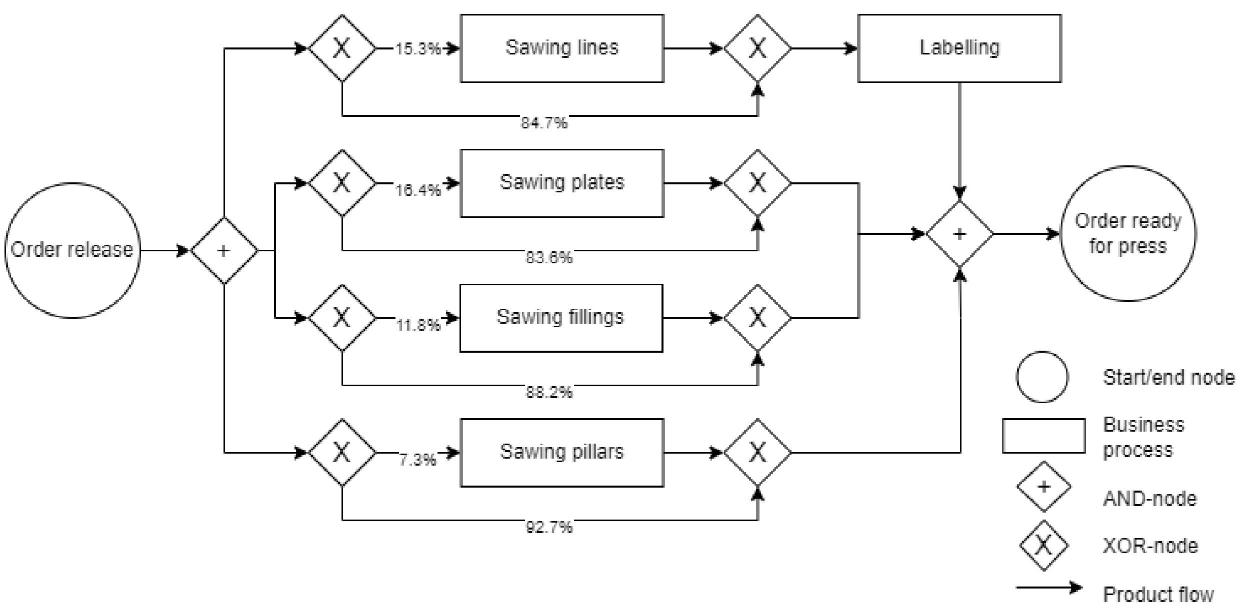


Figure 22: Business Process Model of the sawing and labelling operation

4.6.3 Order ready for press until order ready for transport

When the order is ready for the press, the process becomes a hybrid flow shop, which means that the order can be processed by one machine simultaneously. The order follows a path through the HFS, and all orders go through the presses and the edge processing operations. Not all orders go through the drilling and CNC operations, but all of them do end up in the painter. Part of the orders then go to an external operation, part of the orders go to the SPEC installation, and part of the orders go to the packer. The paths and the percentages of the orders that follow a specific path is visualised in Figure 23. These percentages are determined by analyzing the routes of the orders that have flown through the LDF in the period 1-11-2021 until 31-10-2022. An adjustment for the change in the drill process has been made, which resulted in a lower inflow in the CNC/lock installation operation. Together with stakeholders, it is estimated that approximately 10% of the products will go to the CNC operation, and the remaining percentage of the analysed flow from drill to CNC will go to the Painter directly.

In Figure 23 an external operation is mentioned which has not been mentioned before. Berkvens produces black doors which cannot be processed internally, and therefore part of the process needs to be outsourced. The products that need this external operation are gathered and sent out on Tuesdays and on Thursdays a week later they will return, so they are gone for six workdays. It is important to keep in mind this external operation when determining the sequence, because if a product is ready for the external operation on Wednesday, it has to wait for Tuesday the next week to be sent out. Therefore, the makespan of the door can increase significantly when it has to undergo an external operation.

From Figure 23 it can be seen that some products flow back to the press operations or flow back into the process where the started instead of moving forward in the process. For example, 2.5% of the products that flow out of the painter immediately re-enter the painting operation, and 0.9% of the product flow from the painter back to the press operations. This is due to failures and rework, as sometimes products need to be discarded and produced from scrap, and sometimes products need to be repaired and undergo a specific operation twice, as described in Section 4.6.1. Besides rework, some products need a certain operation multiple times, for instance 5.0% of the products flow from the drilling operation to the drilling operation. Part of this is due to rework, but some doors apparently need multiple drilling operations and hence flow through the Drill operation multiple times.

4.6.4 Main routes

When looking at Figure 23, it seems like all products flow randomly through the system. This is however not the case, as many products follow similar routes. Especially standard products, the MTS products or A-doors, follow a similar route every time, but also MTO products often follow similar routes. To reduce the element of randomness in the flow and to represent reality better, some main routes are determined. These are routes through the system that are encountered often i.e., more than 1% of the products follow a specific route. There are seven main routes, which are assigned a letter from A to G. The main routes are given in Table 9. In total, these routes account for approximately 94.3% of the products that flow through the LDF.

Following a main route means that the general transition probabilities as displayed in Figure 23 are disregarded. This does however not mean that the probability of rework or repairs as stated in Table 8 changes. A specific route does not change the probability of failures, and thus these probabilities should be taken into account when determining the flow through the system.

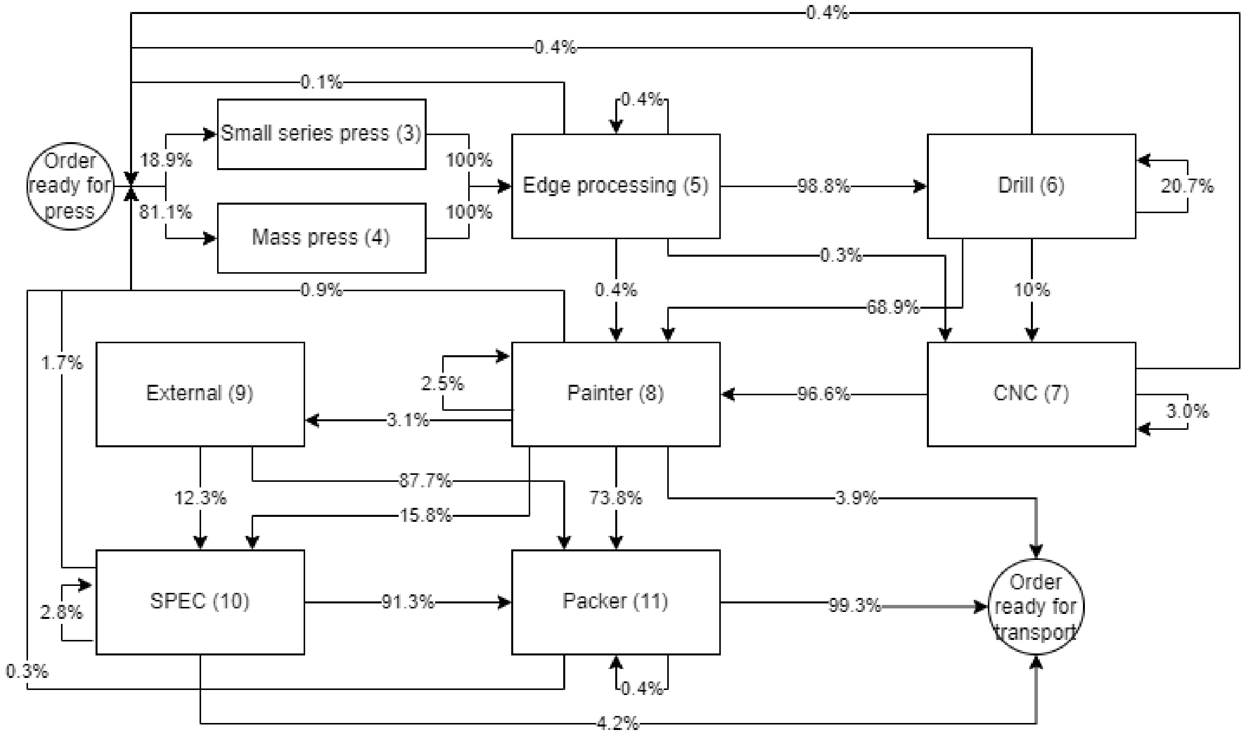


Figure 23: Transition diagram of the Hybrid Flow Shop after labelling

Name	Route							Percentage
A	Edge Processing	Drill	Painter	Packer				56.9%
B	Edge Processing	Drill	Drill	Painter	Packer			14.9%
C	Edge Processing	Drill	Drill	CNC	Painter	SPEC	Packer	11.4%
D	Edge Processing	Drill	CNC	Painter	SPEC	Packer		4.0%
E	Edge Processing	Drill	Painter					3.7%
F	Edge Processing	Drill	Drill	Painter	External	Packer		1.8%
G	Edge Processing	Drill	Painter	External	Packer			1.5%

Table 9: Main routes through LDF

4.6.5 Arrival rate per operation

The daily arrival rate into the system has been determined in Section 4.5. The arrival rates per operation may differ from this, as rework and multiple flows through the same operation should be taken into account. When looking at Table 7, it can be seen that the arrival rate for the total products with the Gamma distribution results in the highest P-value, which means it has the highest fit. When looking at the transition diagrams in Figure 22 and 23, the flows are represented in products, where no distinction is made between MTO and MTS orders. Therefore, the total arrival rate of products into the HFS with a Gamma distribution is chosen as outside arrival rate, i.e. $ProductArrivals \sim \Gamma(\alpha = 8.3, \beta = 270.0, \mu = -711.9)$.

A special case of the Gamma distribution is the Erlang distribution, which is a Gamma distribution with a positive integer α. The Erlang distribution measures the time until the nth event of a Poisson process, in other words: the Erlang distribution is the sum of α independent Exponentially distributed random variables with a mean of β [Montgomery and Runger, 2018] [Mann, 2016]. The

Exponential distribution has the memoryless property, which means that the time until the next event is independent of the previous event. In other words: the time of arrival x is independent of the previous arrival $x-1$. Many arrival processes are characterized by an Exponential process, as the arrival of customers, or orders from customers, is independent of what other customers do. Because of this, it is checked whether the arrival process of products can also follow a Gamma distribution with an integer α. The closest integer to the current α is 8. In order to have the same expected value as before, the scale parameter β should be adjusted as well. The expected value is calculated as follows: $E(x) = \alpha \cdot \beta + \mu = 1527.42$ with the old parameters from the Gamma distribution. β can thus be calculated as follows: $\beta = \frac{E(x)-\mu}{\alpha} = \frac{1527.42+711.94}{8} = 279.92$. The goodness-of-fit test is carried out again with $\alpha = 8, \beta = 279.92$, and location parameter $\mu = -711.9$. The resulting P-value is 0.9529, which means that it is safe to assume $ProductArrivals \sim \Gamma(\alpha = 8, \beta = 270.0, \mu = -711.9)$, which results in an expected value of 1527.42 product arrivals per day.

From Figure 23 it can be seen that the arrivals per operation are not only from outside, but also from the other operations. To determine the arrival rates of all the operations, a system of linear equations can be solved. These equations are based on the premise that everything that flows into a process also flows out of the process, i.e. total arrival rate into server i = outside arrival rate into server i + internal arrival rate into server i. In Figure 23 the operations are assigned a number, furthermore, sawing is considered one operation which is assigned number one and labelling is assigned number two. The system of equations to determine the arrival rates per operation is stated in Equation 25 until Equation 35 in Appendix 3. This system of equations is solved using Matlab. The results are presented in Table 10.

i	1	2	3	4	5	6	7	8	9	10	11
$E(x_i)$	1560.9	1560.9	295.0	1265.9	1567.2	1952.5	206.1	1594.5	49.4	265.4	1468.3

Table 10: Daily arrival rates per operation

4.6.6 Utilization rate of the operations

With the arrival rates determined in the previous section, the utilization rate of the operations can be determined by comparing the arrival rate and production rate of the operations. This is a done to determine whether the system is stable: when the utilization rate of a machine is greater than or equal to 1, the system is unstable. This means that more products need to be processed than the machine capacity can handle. A utilization rate close to 1 will disrupt the system majorly, as the smallest amount of variation will cause the waiting times and the lead time to increase drastically. Normally, the utilization rate of machines is in the 70% - 80% range [Hopp and Spearman, 2011b].

The utilization rate of the operations from Mass press until the Packer is given in Table 11. First, the daily arrivals are taken and the hourly arrivals are calculated by dividing the daily arrivals with the number of hours the operation is running. This is 24 hours per day for the CNC and the Packer, and 16 hours per day for the other operations. When determining the production rate per hour, the standstills determined in Section 4.4 should be taken into account. This is done by taking the production rates determined in Section 4.3 and subtracting the average duration of a standstill with the percentage of days there is a standstill. This gives the hourly production rate with standstills. Lastly, the utilization rate is calculated by dividing the hourly arrivals by the adjusted hourly production rates.

From Table 11 it can be seen that the utilization rate is smaller than 1 for all operations, which means that the system is stable. For some of the operations, the utilization rate is higher than

0.8, which indicates that variance in the system could result in major disruptions in lead time and service level. Besides, the system is utilized well, as none of the utilization rates for the operations is very low, with 0.701 the lowest utilization rate. This means that the machine capacity is utilized well, although there is a risk of major disruptions due to a little variance.

Operation	Daily arrivals	Hourly arrivals	Standstill percentage	Duration standstill	Hourly prod.	Hourly prod. w standstills	Utilization rate
MP	1265.9	79.1	2.2%	8.0	113.1	112.9	0.701
SSP	295.0	18.4	5.3%	2.1	26.0	25.9	0.712
EP	1567.2	97.9	33.8%	2.4	131.6	130.8	0.749
Drilling	1952.5	122.0			137.6	137.6	0.887
Locks/CNC	206.1	8.6	7.1%	0.9	9.9	9.8	0.877
Painter	1594.5	99.7	26.2%	4.1	129.3	128.2	0.777
External	49.4	3.1					
SPEC	265.4	16.6			19.2	19.2	0.862
Packer	1468.3	61.2	20.9%	1.2	73.7	73.4	0.833

Table 11: Utilization rates mass press until packer

Determining the utilization rate of the Sawing and Labelling operations is a little different than for the other operations. As described in Section 4.1.1, there are multiple saws that are used for multiple sawing operations. The rates that are used for determining the sawing and labelling utilization rates are shown in Table 12. The arrival rates for the sawing operations are determined by multiplying the percentages stated in Figure 22 with the total arrival rate into the sawing operation, which is also the arrival rate for the labelling operation. The hourly arrival rates are determined by dividing the daily arrivals by 16 hours. The hourly production is determined in Section 4.3.2.

In Table 13, the types of saws and the number of saws per type are stated. The hourly arrival rate for the shortening saw is the sum of the arrival rates for Sawing Lines and Sawing Pillars, which both have to be processed by the shortening saw. The average production rate for the shortening saw is calculated by multiplying the proportion of products of a certain operation with the production rate. Lastly the utilization rate is calculated again by dividing the hourly arrival rate by the hourly production rate. The utilization rate for the sawing operations is very low. This can be explained by the fact that these saws are also used for the production of the HPL doors. As only data about the production of the lacquered doors is gathered, the true utilization rate of the saws will be higher than stated in this table.

Operation	Daily arrivals	Hourly arrivals	Hourly production
Sawing Fillings	226.4	14.2	28.8
Sawing Lines	242.7	15.2	107.1
Sawing Plates	174.6	10.9	93.1
Sawing Pillars	108.0	6.8	117.0
Labelling	1480.0	92.5	134.6

Table 12: Arrival and production rates sawing and labelling

Machine	Nr. of saws	Hourly arrival rate	Hourly production rate	Util. rate
Shortening saw	3	21.9	110.2	0.066
Plate saw	1	10.9	93.1	0.117
Fillings saw	2	14.2	28.8	0.246
Labelling	-	92.5	134.6	0.687

Table 13: Utilization rates sawing and labelling

4.7 Chapter conclusion

In this chapter, the current performance of the operations in the flow shop is analysed and the first sub research question is answered, which was: "What are the parameters and performances of the individual operations and how do the operations interact with and influence each other?". Data has been gathered and is analysed, which resulted in the determination of the parameters of the individual operations. For each individual operation many things have been analysed, namely: the processes within the operation, the production rates and underlying distributions, the standstills of the machines, and the probability on rework and repair. Furthermore, the arrival process has been analysed as well. Besides analysis on the individual operations, also the interaction between the operations has been analysed. The flow through the factory is determined and the main routes and transition probabilities are determined. Based on these routes and transition probabilities, the arrival rates per operation could be determined and compared with the production rates. Considering all relevant factors, such as standstills and rework, the utilization rates could be determined. Based on these utilization rates it can be concluded that the LDF of Berkvens should be able to deal with the demand based on the performance of the operation, as the utilization rates of the individual operations are well below 1. In the next chapter, more information will be gathered about the processes that influence the performance of the LDF.

Besides answering the first sub research question, important analyses for the second research question have been done as well, which was: "What distinction can be made between MTO and MTS orders regarding current performance and required performance, and how should the differences between those be addressed in the sequencing of orders?". The difference in arrival rate between MTO and MTS orders has been investigated, and in can be concluded that the average size of MTS orders is much bigger than MTO orders. Unfortunately, the analyses of the arrival rates for the MTO and MTS orders produced to significant results, meaning that an individual arrival process for both the MTO and the MTS orders cannot be defined. Thus a general arrival process should be used for further investigations to answer this sub-research question.

5 Current sequencing and planning procedure

In this chapter, the current sequencing and planning procedures will be investigated. Two sub research questions have a central role in this chapter. The third SRQ has a central role, which is: "How is the due date for production orders determined and what adjustments can be made in this based on lead times?". The second theme that is central is based on the first part of the fourth SRQ, namely: "How is the production sequence of jobs and batches determined ...?". In this chapter, first the current sequencing procedure is investigated. Then the batching method is investigated, followed by the determination of the theoretical lead time. Lastly, the procedure to determine the due date is investigated.

5.1 Current sequencing procedure

Currently, determining the sequence in which products are produced is mainly done manually. The sales orders are known and an recommendation for production and procurement of materials is generated by the ERP-system. Those recommendations are manually judged and changed if deemed necessary, after which they are recorded. The production orders are made based on these recommendations. The due date of the production order, which is the internal due date, is determined based on the due date of the sales order, which is the external due date. The date on which the order is released is determined by having an estimated lead time per operation that a product needs and summing these estimations. Then the number of days based on the estimations is subtracted from the internal due date, which gives the order release date. When the number of sales orders for a certain period is small, orders for MTS doors are also generated and their release dates are determined, so these can also be produced.

The production orders are batched, which will be explained later in more detail. When the order release date is known and reached, the lists with the batches and raw materials are printed and send to a foreman, who gathers all the batches that need to be produced. This foreman is mainly in charge when determining what batches are released for production first, where he considers the sizes of the buffers, the number of employees, the maintenance, etc. He does this based on experience and some logic. The pick-orders for the raw materials are given to the operators by the foreman, which results in the start of production.

Each batch has its own order type, which can be rush, New Logistic Concept (NLC), France, or stock replenishing. Doors are transported in stacks with a maximum of 24 doors per stack within the facility. This is done by rolling the stacks of doors on roller conveyors, where the doors need to be pushed manually. Stacks are moved to different conveyors with a transportation device that needs to be handled manually as well. A coloured marker is added to a stack to indicate a higher priority for this stack. The operators at each operation have a screen where they can see what orders need to be produced each day, and they can choose for themselves which orders they produce first. However, they need to abide the priority rules, so rush orders are produced first, then NLC, etc.

5.2 Batching method

Products flow through the facility in multiple forms. In the sawing and labelling operations, one door consists out of multiple individual materials: the pillars, lines, fillings, and plates. All those individual parts are assembled together by the presses to make the trunk of the door. When the press is finished, the products are moved further through the LDF in batches. The number of products in the batch varies: a batch can consist out of 3 doors, but also out of 60 doors.

The batching script assembles multiple production orders in one batch that flows through the flow shop. A batch consists out of least one stack of doors, but can also contain multiple stacks. The doors that make up the batch should have similar specifications to ensure the most efficient production possible. There are multiple locations inside the factory where batches can be formed. The first time a batch is formed on paper is at the start of the process, before the order release, as described in section 5.1. This batch does not flow through the LDF together physically until the product has been processed by either one of the presses. The batch remains intact until the drilling process, where new batches can be formed as four stacks of doors can be processed simultaneously by this operation. This means that small changes in batches can be made for the further operations. The same is possible at the CNC operations, which can handle two stacks simultaneously.

Doors are batched according to their specifications, which can be various things. For instance, the sizes of the door, the orientation of the door, the model of the door, the colour of the door, the type of hinges of the door, the list goes on. Some operations can handle doors with certain specifications while other specifications don't need to be taken into account. For instance, for the painting line the measurements or the orientation of the door does not matter, while for the drilling line it does not matter what colour the door needs to be. Therefore, the ability of the flow shop to change the composition of the batches ensures bigger batches can flow through the flow shop. This means that on average higher stacks can be built and less handling time and setup time is required for all the batches.

In Section 4.6.1 it has been investigated that some doors need rework and repair. When a door needs to be discarded totally, it gets appended to a new batch and is produced from scrap. When a door needs to be repaired, mostly it stays assigned to the same batch as before. It may happen that the batch flows on through the system and that the one door that needed reparation is put on a separate stack and flows through the facility individually. However, it may also be the case that the door is repaired before the rest of the batch flows through the next operation. Then the door is appended physically to the batch again, and no additional stacks should be built.

5.3 Theoretical lead time

The lead time of doors is estimated roughly to determine the order release date, which is the theoretical lead time. This estimation is a rough estimation and includes much slack for waiting times and other process disruptions. Lead time estimations are done in workdays, so work that is done on Saturday is a bonus. A distinction can be made between the three types of doors, which are A-, B-, and C-doors, where A are MTS doors and B and C are MTO doors.

Estimated lead times for the MTS doors are fairly constant, which is logical as this type of door is standard and is kept in stock. The production of orders that consist out of MTS doors is intended to increase the inventory. Berkvens works with a safety inventory that can handle sudden increases in demand, and therefore MTS orders always have the same priority level and estimated lead time. The lead time for MTS products consists out of estimated times for the individual operations and a safety factor. The estimated lead time for an MTS order is stated in Table 14, where the times are in workdays. From this table it can be seen that the total lead time for the A-doors is quite long, which can be explained by the fact that there is normally no rush for completing A-doors, as these are mainly used to fill up capacity. A notable thing is that there is no time reserved for the drilling operation, which is explained by the fact that this time is incorporated in the time for edge processing and painting.

Estimated lead times for the MTO door of type C is fairly constant as well. The estimated lead time for the individual operations is the same for all the doors and is given in Table 14 again. This lead time has more variance than the lead time for the A-doors, as these doors are MTO which means that the complexity of the door might change, which may require additional time for certain operations. The time mentioned in the table is the most common time reserved for C-Doors. From this table it can be seen that many C-doors have specifications that often need some time for the SPEC operation, which is logical because MTO doors are in general more complex than A-Doors.

The estimated lead time for the MTO door of type B varies mostly. There are a number of standard door types for which the theoretical lead time is fairly constant. Those are mentioned in Table 14 for the NLC door, France door type 1, and France door type 2. The NLC door has a standard external lead time of 6 days. A door should meet certain specifications before it can be delivered by NLC, such as a certain colour, size, and no specifications that require the CNC operation. Similarly, a door should meet certain specifications before it can be a France type 1 door or a France type 2 door. A lot of doors do not fall in any of these categories, and their theoretical lead time completely depends on the specifications desired by the customer.

The theoretical lead time is calculated by the ERP-system, which also gives an advice for the moment of order release. It should be noted that this is only an advice, which means someone should agree to this advice or change it. This is done because the ERP-system fails to take the actual situation of the factory into account. For example, if major maintenance is planned for a certain machine, this might mean that the order is stuck for an additional day in the buffer, waiting for the machine to be operational again. This additional day is taken into account when the order is released by an actual person. The ERP-system does however not know this planned maintenance, so it cannot take that additional day into account. There might be other situations where the actual order release date deviates from the advised order release date, such as production stops due to holidays, delayed release because of a busy period or early release due to a quiet period, etc.

Operation	**A-doors**	**C-Doors**	**B-Doors**		
			NLC	**France 1**	**France 2**
Sawing+Labelling	5	1	1	1	1
Press	5	1	1	1	2
Edge processing	2	1	1	1	2
Drilling		1		1	2
CNC					2
Locks		0.5		0.5	1
Painting	3	1	2	1	7
SPEC		1.5		0.5	1.5
Packing	5	1	1	1	1
Total	**20**	**8**	**6**	**7**	**19.5**
Lead time	**20**	**8**	**6**	**7**	**20**

Table 14: Theoretical lead times

5.4 Determining the due date

The due date of orders is set when a sales order is realized. The due date is normally eight weeks after the sales order is made. When a sale is realized, the order processing is started immediately and the first step is to check the material requirements and purchase materials for the order if necessary. For most orders, the production order is released five weeks before the product is due, to ensure the order is ready for delivery on its due date. Some orders deviate from this standard procedures, because raw materials may not be present yet, or because other special circumstances.

Deviations from this standard approach may occur. For NLC orders, the standard delivery time is 2 weeks, thus the due date is set two weeks after the sales order is realized. The NLC products have a higher priority, which means that they should flow faster through the factory. Within the NLC products, two different priority levels should be distinguished. There are two large customers that order approximately 50% of the NLC doors that have a higher penalty clause in their contracts than other NLC customers, so the orders of these two customers should be prioritized. Besides the deviation from the standard due date determination for NLC customers, the due date of some orders can be determined much further in advance. When a customer has a very large project that takes several months or years to complete, for instance building a complete hotel or apartment building, that customer knows far in advance it will need a large number of doors. This means that the due date will be known far in advance as well, although this due date might shift later.

Shifts in due dates occur quite frequently, in approximately 20-25% of the orders the due date is shifted to a later moment. Reasons for this shift are mainly disruptions and delays in the building process, for instance due to the weather conditions or shortages of material. Installing doors is most of the time one of the last steps of the building process, which means that all disruptions that occur influence the process. Customers communicate disruptions with the sales representatives of Berkvens, which in turn communicate this with the LDF. Changes in due date may happen far in advance, but may also happen when the production order for the doors is already released, thus the production of the doors has already started. When this happens, the finished doors are stored in the warehouse.

5.5 Chapter conclusion

In this chapter, the current sequencing and planning procedures are investigated. The third SRQ was: "How is the due date for production orders determined and what adjustments can be made in this based on lead times?". The due dates for standard MTO orders are determined when the sales order for the doors are placed. Normally, the production order is released five weeks before the external due date. The due dates for NLC orders are normally two weeks after the placement of accompanying sales order. The determination of the due dates is mostly done based on experience and general estimates: from past experience the approximate time that is needed for completely producing doors is estimated, and the release date of production orders are based on this. The estimates for the production time needed are very general and mostly independent of the type of door that is ordered. If better and more reliable estimates would exist on the production and waiting time per operation for each type of door, release dates could be better estimated. This would lead to an decreased work-in-process inventory in the production facility, which in turn would lead to a smoother flow of products through the factory. Unfortunately, at the moment these estimates are not yet available, but in the future these might become available. When this becomes reality, new and better estimates for the lead times should be established.

The first part of the fourth SRQ was: "How is the production sequence of jobs and batches determined...?". Currently, the sequencing of jobs and batches is mainly done by the operators at the machines. Jobs are assigned to batches based on specifications and the route of the doors. Each batch consists out of one or more stacks of doors that stick together during production. Every batch has a priority, and the operators should adhere the priority rules. In short this means that first the rush batches should be done, then NLC, then France, and lastly MTS orders. If there are multiple batches with the same priority level, the operators pick the next batch themselves.

6 Mathematical model for sequencing problem

In the literature review conducted as a start of this thesis [Jeuken, 2022], it has been concluded that it is helpful to build a mathematical model to represent the relations between various variables and to understand the working of the LDF better. Thus, a mathematical model is built and presented here. The objective of the mathematical model is to minimize the number of tardy jobs and the makespan of all jobs, which is based on the second part of the fourth SRQ: "... what improvements can be made in the sequencing with the objective to minimize the tardiness and makespan of orders?". This mathematical model has elements of mathematical models presented in [Tan et al., 2018], [Ünal et al., 2020], [Ruiz and Vázquez-Rodríguez, 2010], [Voß and Witt, 2007], and [Bernate Lara et al., 2013].

6.1 Parameters

In Table 15 the sets, input parameters, output parameters, and decision variables of the mathematical model are given, which are used in the objective function and the constraints. There are four sets used in the mathematical model. Set J, B, and M are straightforward; they represent all jobs, batches, and machines respectively. Set M_{batch} is a subset of set M: this set contains all the machines where batch production is used. There are only two decision variables in the mathematical model, namely the start time of a job on every machine, and the assignment of a job to a specific batch for a specific machine.

6.2 Objective function and constraints

Below, the objective function and the constraints of the mathematical model are given and explained.

Equation 1 gives the objective function for this mathematical model. The objective is to minimize the total makespan F_j and the number of tardy doors T_j in the LDF. Weights (w_1 and w_2) are added to the two factors, to balance them out and to indicate where the priority lies.

$$Minimize \qquad \sum^{J}(F_j \cdot w_1 + T_j \cdot w_2) \tag{1}$$

Equation 2 ensures that the processing of a job on any machine $t_{m,j}$ can only start after the job's release time r_j.

$$t_{m,j} \geq r_j \qquad \forall m \in M, \forall j \in J \tag{2}$$

Equation 3 ensures that a product is only added to a batch for a machine (decision variable $X_{m,j,b}$ if it needs to be processed by that machine according to its route $Z_{m,j}$. Furthermore, it also ensures that the product is only assigned to one batch.

$$\sum_{b=0}^{B} X_{m,j,b} = Z_{m,j} \quad \forall m \in M \forall j \in J \tag{3}$$

Equation 4 states that the processing time of a batch $p_{m,b}$ is the sum of the processing times of the jobs $p_{m,j}$ that are assigned to that batch (with decision variable $X_{m,j,b}$. This only holds for machines where production is done in batches.

$$p_{m,b} = \sum^{J} p_{m,j} \cdot X_{m,j,b} \qquad \forall m \in M_{batch}, \forall b \in B \tag{4}$$

Sets	
J	Set of jobs
B	Set of batches
M	Set of machines
M_{batch}	Subset of machines where production is batch based, $S_b \subseteq S$
Input parameters	
r_j	Release date of job $j \in J$
d_j	Due date of job $j \in J$
$Z_{m,j}$	Binary variable: 1 if job $j \in J$ should be processed by machine $m \in M$
$p_{m,j}$	processing time of job $j \in J$ on machine $m \in M$
f_m	Flowtime of one product in machine $m \in M_{batch}$
K_m	Buffer capacity of machine $m \in M$
D	Max batch size for any batch on any machine
w_i	Weight to prioritize an element of the objective function, $i = 1, 2$
Output parameters	
$c_{m,b}$	Completion time of batch $b \in B$ on machine $m \in M_{batch}$
C_b	Completion time of batch $b \in B$ in last stage
C_j	Completion time of job $j \in J$ in last stage
T_j	Binary variable, 1 if job $j \in J$ is tardy, 0 otherwise
L_j	Lateness of job $j \in J$ (may be negative)
F_j	Makespan of job $j \in J$
$H_{m1,m2,b}$	Binary variable: 1 if batch $b \in B$ flows directly from $m1 \in M_{batch}$ to $m2 \in M_{batch}$
$e_{m,j}$	End processing time of job $j \in J$ on machine $m \in M$
$t_{m,b}$	Start processing time of batch $b \in B$ on machine $m \in M_{batch}$
$p_{m,b}$	Processing time of batch $b \in B$ on machine $m \in M_{batch}$
$e_{m,b}$	End time (when batch is available for next machine) of batch $b \in B$ on machine $m \in M_{batch}$
$O_{m,b1,b2}$	Binary variable, 1 if batch $b1 \in B$ is processed before $b2 \in B$ on machine $m \in M_{batch}$
$L_{m,b}$	Number of products batch $b \in B$ consists out of when flowing through machine $m \in M_{batch}$
$S_{m,b}$	Number of stacks batch $b \in B$ consists out of when flowing through machine $m \in M_{batch}$
$N_{m,b1,b2}$	Binary variable: 1 if batch $b1 \in B$ and batch $b2 \in B$ are simultaneously in the buffer before machine $m \in M_{batch}$
Decision variables	
$t_{m,j}$	Time at which job $j \in J$ starts on machine $m \in M$
$X_{m,j,b}$	Binary variable: 1 if job $j \in J$ is assigned to batch $b \in B$ when processed on machine $m \in M_{batch}$

Table 15: Model sets, parameters, and decision variables

Equation 5 enforces that the end time of a batch $e_{m,b}$ is the start time of said batch $t_{m,b}$ plus the processing time $p_{m,b}$ plus the flow time f_m. The flow time should be added, as the processing time only takes the time products need on machine into account, while the time the products need to flow through the machine is disregarded. Therefore, this time should be added to ensure the production on the next machine does not start before the batch has flowed out of the previous machine.

$$e_{m,b} = t_{m,b} + p_{m,b} + f_m \qquad \forall m \in M_{batch}, \forall b \in B \tag{5}$$

Equation 6 and 7 ensure that the start and end time of a job ($t_{m,j}$ and $e_{m,j}$) are the same as the start and end time of the batch ($t_{m,b}$ and $e_{m,b}$) they are assigned to (with decision variable $X_{m,j,b}$).

$$t_{m,j} = \sum^{B} X_{m,j,b} \cdot t_{m,b} \qquad \forall j \in J, \forall m \in M_{batch} \tag{6}$$

$$e_{m,j} = \sum^{B} X_{m,j,b} \cdot e_{m,b} \qquad \forall j \in J, \forall m \in M_{batch} \tag{7}$$

Equation 8 ensures that the start time of processing a job $t_{m,j}$ should be bigger than the end time of the processing time of that job on any previous machine $e_{m-i,j}$ where batch production is used. This is thus a precedence constraint, i.e., this constraint ensures that the job visits the operations in the right order. For the machines that do not produce in batch, the sawing and labelling operations, this constraint is unnecessary, as the product can be split up into multiple production orders and can thus be processed by several machines simultaneously.

$$t_{m,j} \geq e_{m-i,j} \cdot Z_{m-i,j} \qquad \forall i \in [1, m], \forall m \in M_{batch}, \forall j \in J \tag{8}$$

Equation 9 states that the completion time of a job C_j is the maximum of the end times of the job in all machines $e_{m,j}$.

$$C_j = MAX[e_{m,j} \quad \forall m \in M] \qquad \forall j \in J \tag{9}$$

Equation 10 states that the job is tardy T_j when the completion time of the job C_j is greater than the due date of the job d_j. The objective is to minimize the number of tardy jobs, and therefore the tardiness is expressed in a binary variable.

$$T_j = \begin{cases} 1, & \text{if } C_j > d_j \\ 0, & \text{otherwise} \end{cases} \qquad \forall j \in J \tag{10}$$

Equation 11 determines whether batch $b1$ is produced before batch $b2$ on machine m with the binary variable $O_{m,b1,b2}$. This is necessary for Equation 12, which ensures that the production of a batch $t_{m,b}$ can only start when the machine is available, which is when the batches preceeding this batch are done with the production. In Equation 5 the flow time is taken into account, which should not be done here, as the production of the next batch can already start when the previous batch is still flowing through the end stages of the machine.

$$O_{m,b1,b2} = \begin{cases} 1, & \text{if } t_{m,b1} < t_{m,b2} \\ 0, & \text{otherwise} \end{cases} \qquad \forall m \in M_{batch}, \forall b1 \in B, \forall b2 \in B \tag{11}$$

$$t_{m,b2} \geq (t_{m,b1} + p_{m,b1}) \cdot O_{m,b1,b2} \qquad \forall m \in M_{batch}, \forall b1 \in B, \forall b2 \in B \tag{12}$$

Equation 13 states that the makespan of a job F_j is its completion time C_j minus its release time r_j.

$$F_j = C_j - r_j \qquad \forall j \in J \tag{13}$$

Equation 14 until 16 ensure that the products that make up a batch in a certain operation, stay in that batch until new batches can be made, on the condition that the products in that batch need to be processed by a certain operation with the decision variable $X_{m,j,b}$. New batches can be made at the drilling operation (operation 8) and the CNC (operation 9).

$$X_{5,j,b} \cdot Z_{5,j} = X_{7,j,b} \cdot Z_{7,j} = X_{8,j,b} \cdot Z_{8,j} \qquad \forall j \in J, \forall b \in B \tag{14}$$

$$X_{6,j,b} \cdot Z_{6,j} = X_{7,j,b} \cdot Z_{7,j} = X_{8,j,b} \cdot Z_{8,j} \qquad \forall j \in J, \forall b \in B \tag{15}$$

$$X_{10,j,b} \cdot Z_{10,j} = X_{11,j,b} \cdot Z_{11,j} = X_{12,j,b} \cdot Z_{12,j} \qquad \forall j \in J, \forall b \in B \tag{16}$$

Equation 17 counts the number of products that are assigned to a particular batch $L_{m,b}$. Equation 18 ensures that the number of products assigned to a batch does not exceed the maximum batch size D. Equation 19 then determines the number of stacks $S_{m,b}$ the batch consists of by dividing the number of products in a batch by the maximum stack height, which is 24. This number is rounded up, as a batch of one consists out of one stack, but a batch of 24 products also consists out of one stack.

$$L_{m,b} = \sum^{J} X_{m,j,b} \qquad \forall m \in M_{batch}, \forall b \in B \tag{17}$$

$$L_{m,b} \leq D \qquad \forall m \in M_{batch}, \forall b \in B \tag{18}$$

$$S_{m,b} = roundup(L_{m,b}/24) \qquad \forall m \in M_{batch}, \forall b \in B \tag{19}$$

Equation 20 ensures that the binary variable that checks whether batches are simultaneously in a buffer $N_{m,b1,b2}$ is right. This is done by checking whether the end time of a batch ($b2$) in any previous machine ($m-i$)is smaller than the start time of any other batch ($b1$) if this batch flows directly from a previous machine to the machine that is considered here, i.e., if $H_{m-i,m,b2} = 1$.

$$N_{m,b1,b2} = \begin{cases} 1, & \text{if } e_{m-i,b2} \cdot H_{m-i,m,b2} < t_{m,b1} \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in [0,m], \forall b1 \in B, \forall b2 \in B, \forall m \in M_{batch} \tag{20}$$

Equation 21 ensures that the capacity of any buffer is not exceeded. This is done by checking what batches are in a buffer simultaneously $N_{m,b1,b2}$ and adding up the number of stacks of these batches $S_{m,b}$, which should be smaller or equal to the capacity of the buffer K_m.

$$S_{m,b1} + \sum_{b2=0}^{B} N_{m,b1,b2} \cdot S_{m,b2} \leq K_m \qquad \forall b1 \in B, \forall m \in M_{batch} \tag{21}$$

6.3 Solving the mathematical model

In the literature review done as preparation on this thesis [Jeuken, 2022], exact methods to solve the hybrid flow shop scheduling problem, such as the mathematical model above, have been reviewed. It has been concluded that this type of problems is NP-hard, which means that the problem is inherently intractable, that no algorithm can solve it quickly, and that the problem is 'just as hard' as a problem that is recognized by experts as being difficult. This means that an exponential amount of time might be needed to discover a solution [Gupta, 1988], and that the solution itself is so extensive that often it cannot be described with an expression of the length bounded by a polynomial function of the input lengths [Mousavi et al., 2013]. In short, an NP-hard problem might not be solved in polynomial time [Garey and Johnson, 1979]. The mathematical model described above is also NP-hard, which means it might take an exponential amount of time to solve this problem. In this problem, the number of jobs that need to be scheduled is fixed, which means that every time a new job arrives, or a job needs to be repaired or discarded, the model should be solved again. It is thus impossible to base the sequence on the mathematical model above, as the amount of time the model needs to find an optimal solution surpasses the amount of time between the need for two solutions. In other words: a new optimal solution is already required before the previous optimal solution is found. This explains the need for heuristics and metaheuristics, as these find an optimal solution much faster.

The mathematical model above has been created in Python with the help of the Gurobi library [Gurobi Optimization, LLC, 2023]. The Gurobi library has been chosen, as the author is familiar with the software, Gurobi is the market leader in mathematical optimization software, and it offers free academic licenses to students. Gurobi uses the branch-and-cut algorithm to solve the mathematical model above, which is a branch-and-bound algorithm that uses cutting planes to tighten the boundaries. This means that many iterations are done to find the most optimal solution. Every iteration explores another 'branch' of the 'tree' of possible solutions.

Even though the mathematical model is NP-hard, the model can be used to solve a small set of products to make conclusions about the performance of the model. A set of 10 products has been created, with their corresponding routes, due dates, release dates, and service times. These input parameters are loaded into the model and the model is executed. As can be expected, none of the products in the small data set were tardy, as the system is almost empty, thus products have almost no waiting time. Therefore, the value of w_1 and w_2 does not have any influence on the model, as the tardiness of every job (T_j) is 0. The makespan of every job (F_j) is thus the only parameter that influences the objective function.

In Table 16 the results of decision variable $X_{m,j,b}$ are summarized. Here, the number of the batch that a product is assigned to is shown for every operation where products are processed in batches. The results displayed here are easily explained; products 1 until 5 are released on the first day and all follow the same path, and hence they are processed in the same batch. Products 6 until 10 are released on the second day, and the routes of the products vary more, as product 8 should be processed by the small series press (SSP), the CNC, and in the SPEC department, which causes the need for this product to have a separate batch. Furthermore, product 10 needs an external operation, and is thus batched separately from the other products after the drill, where batches can be made again.

In Table 17 the results of the decision variable $t_{m,j}$, as well as the output parameters C_j and F_j are given, where the times are in seconds and the first day starts at time 0. From this table it becomes clear that the products need to wait for all products in the batch before they can start processing, and all products that are not batched together need to wait for the operation to become available again. The completion time C_j is the same for all products that are in the same batch at the packing operation. The makespan F_j is the completion time minus the release time. The total makespan, and thus the objective values is 627,922 seconds, which is equal to 174.4 hours. A large proportion of this makespan is contributed by the time for the External operation, which took 6 workdays in total. The results in this table are as expected: the products flow in batches as long as possible, and only when changes in batches are required, products are split up. It took several minutes to solve this model by gurobi, due to the large solution space. Remember: Table 16 gives a summary of 10 matrices, so the full solution is a much more elaborate expression.

Product	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
SSP								3		
MP	1	1	1	1	1	2	2		2	2
EP	1	1	1	1	1	2	2	3	2	2
Drill	1	1	1	1	1	2	2	3	2	2
CNC								3		
Painter	1	1	1	1	1	2	2	3	2	4
External										4
SPEC								3		
Packer	1	1	1	1	1	2	2	3	2	4

Table 16: Result $X_{m,j,b}$ mathematical model

Product	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
sFillings									86,400	
sLines										
sPlates				0		86,400		86,400		
sPillars										
Labelling	0	23	0	53	102	86,421	86,453	86,466	86,400	86,400
SSP								86,589		
MP	175	175	175	175	175	86,471	86,471		86,471	86,471
EP	2,121	2,121	2,121	2,121	2,121	88,400	88,400	131,207	88,400	88,400
Drill	2,555	2,555	2,555	2,555	2,555	88,846	88,846	131,527	88,846	88,846
CNC								131,733		
Painter	2,865	2,865	2,865	2,865	2,865	89,131	89,131	131,999	89,131	89,250
External									89,995	
SPEC								132,753		
Packer	3,730	3,730	3,730	3,730	3,730	89,970	89,970	157,953	89,970	608,395
C_j	4,328	4,328	4,328	4,328	4,328	90,452	90,452	158,321	90,452	608,743
F_j	4,328	4,328	4,328	4,328	4,328	4,052	4,052	71,921	4,052	522,343

Table 17: Result $t_{m,j}$, C_j, and F_j mathematical model

6.4 Chapter conclusion

In this chapter a mathematical model for the sequencing problem in the LDF is created, to partly answer the second part of the fourth SRQ: "... what improvements can be made in the sequencing with the objective to minimize the tardiness and makespan of orders?" The developed mathematical model fails to answer this SRQ, as the mathematical model is NP-hard, meaning that no good solution is found quick enough to actually implement. Therefore, heuristics and metaheuristics will be developed and tested in the remainder of this thesis. The next chapter will explain the framework that will be used to test the performance of the (meta)heuristics, which will be described in Chapter 8.

7 Simulation model

From the analysis in Chapter 4 and 5 it becomes clear that the real situation in the LDF of Berkvens is quite complex. There are many rules and exceptions on these rules, and products flow continuously through operations that have varying opening hours and breakdowns. Therefore, the workings of the LDF cannot be expressed easily in a mathematical model, and this model would be hard to solve, as explained in the previous Chapter. Simulation is therefore a better method to determine the performance of the LDF, and can be used to evaluate the current sequencing method and new sequencing heuristics, thus helping in answering the second part of the fourth SRQ: "... what improvements can be made in the sequencing with the objective to minimize the tardiness and makespan of orders?".

"Simulation refers to the imitation of the operation of a real-world process or system over time" ([Boon et al., 2021] p. 3). There are many methods for simulation, and in this research Discrete Event Simulation (DES) is chosen for doing the simulation. A discrete-event system is determined by a sequence of random event times, and every time one of those random event happens, the state of the system changes [Fishman, 2013]. As this is the case in the LDF of Berkvens, DES is a suitable method of simulation.

In the remainder of this chapter, first the elements of the simulation model are explained by describing the main workings of the model. Then the model needs to be verified and validated. After that, the warm-up period, the duration of the simulation, and the number of runs are determined. Lastly, the model is executed for the current situation of the LDF and the results obtained by the model are compared to the analyzed situation.

7.1 Model simulation with Discrete Event Simulation (DES)

The simulation is implemented with an object oriented approach, meaning it consists out of many objects with properties (parameters) and methods (simulating or determining properties). The events which form the basis of the simulation are also objects which have amongst others two important properties, namely the type of event and the time of the event. Events are stored in the Future Event Set (FES), which is a sorted list of scheduled future events. As the state of the system only changes when an event takes place, the first event should be picked from the FES, and based on the details of this event, some actions need to be carried out. Various objects (or classes) are used in the DES, including the Product class, the Batch class, the Event class, the FES class, the NetworkSimulation class, the NetworkSimResults class, and the SimResults class. These classes and their properties and methods are described in Appendix 4. The actual simulation is executed by the *simulate* method in the NetworkSimulation class. This method is quite extensive, as all exceptions and complexities of the LDF are expressed in code here. Therefore, the working of this method is briefly explained below and in pseudo code in the appendix, and the main functions of the simulation are explained.

7.1.1 New arrivals and batching

New arrivals form the basis of the simulation, as new arrivals trigger arrival events, which form the backbone of DES. Every day new products arrive, for which a priority, route, due date, and service times for individual operations are determined. Arrival events are made for the sawing and labelling activities, which can be executed simultaneously in some cases. This process is described in Algorithm 3 in Appendix 5. Furthermore, the products need to be batched, and this process is described in Algorithm 4 in Appendix 5 as well. Here, all products that need to be assigned to a batch are gathered in the *PoolForBatching* list. It is checked whether a batch with certain characteristics already exists, and if so the product is appended to that batch. Otherwise, a new batch is made and the product is appended to that batch. Lastly, the service times for all operations with batch flow is determined and the number of stacks a batch consists of is determined. These parameters are stored in the Batch object.

7.1.2 Product flow

In section 4.6 the relations between the operations are described. From this section it becomes clear that in the sawing and labelling operations the products flow individually, and from the presses on products flow in batches. There are three queues for the sawing operation, one for the shortening saw, one for the fillings saw, and one for the plate saw. There is one more queue for the labelling operation. Arrival events are made based on the route of the product and the event includes information about the location in which the event takes place, which is the queue where the products arrives. The algorithm for an arrival event in a sawing or labelling activities, which are locations 0 until 3, is shown in Algorithm 5 in Appendix 5. First the product is added to the queue, which is then sorted based on priority level. If there is a server available, then the product departure can immediately be scheduled based on the current time plus the service time. A departure event is made and added to the FES.

The algorithm of a departure event in the sawing or labelling activities is shown in Algorithm 6 in Appendix 5. First, the product is removed from the queue and the status of the corresponding sawing/labelling activity is changed to complete. If the event location is 0, which is the sawing lines activity, the labelling activity should be scheduled next, as this can only happen when the lines are finished. Otherwise, a check is done to determine whether all sawing and labelling activities are done, and if this is the case, then it is checked whether all products in the batch are ready for the press. If all products are ready for the press, the waiting time of all products in the batch is registered, the next queue is determined based on the route, which can either be queue 4 (small series press) or 5 (mass press), and the next service time is determined. Then, an arrival event is made for the batch at current time plus the service time, which is added to the FES. If there were products waiting in the queue for the current activity, the next product is selected, its service time determined, and a new departure event is made and added to the FES.

7.1.3 Batch flow

The algorithms for the batch flow are different from the algorithms from the product flow, because the operations that the batches undergo are more complex. Here, the processes are disrupted more often, for instance due to blockages, opening hours, breakdowns, repairs and discards, etc. The algorithm for an arrival event for one of the batch operations is shown in Algorithm 7 in Appendix 5.

First the batch is added to the queue and the queue is sorted based on the highest priority. The number of stacks the batch consists of is added to the buffer, and if the buffer size is smaller than or equal to the number of stacks in the buffer, the previous machine is blocked, such that no more batches can arrive in the buffer. Next, it is checked whether the queue length is smaller than the number of servers and the operation is not blocked due to a full buffer in the successive operation. If this is the case, the batch can be processed immediately, so the waiting time is registered, an departure event is made, and this event is added to the FES. If there are servers available to process the batch, but the operation is blocked due to a full buffer, the batch is added to a separate queue with products that are waiting due to the block.

In Algorithm 1 the pseudo-code for a batch departure is given. As the handling of a batch departure event is quite extensive, the pseudo-code is reported here for clarity. First it is checked whether the machine is blocked due to a full buffer in a successive operation by checking. If this is the case, the batch is added to the list of products that are waiting for the block to be over. Then a check is executed to see if the machine is closed when regarding the opening hours of the machine. If this is the case, the additional waiting time needs to be registered, and a new departure event for the batch should be made and added to the FES. Next, another check is executed, this time to see if the machine has a breakdown. If this is the case, again the additional waiting time needs to be registered, and a new departure event for the batch should be made and added to the FES.

If all the previous check are passed, the batch can actually be processed, so the batch is removed from the queue. There is a chance on product repairs or products discards, which should be determined for every product in the batch individually. If the product needs to be repaired, it is added to the ToBeRepaired list. If the product needs to be discarded, it is added to the ToBeDiscarded list. If there are products in the ToBeRepaired list, the priority of the current batch should be investigated, and if this priority is 2, a new batch instance with priority 1 should be made, with all other batch parameters the same as the batch that is currently processed. If the priority of the current batch is not 2, a new batch instance should be made with the same parameters as the current batch. An arrival event for the new batch should be scheduled for 8 hours later, which is the average time it takes to repair a product, and this event should be added to the FES. If there are products in the ToBeDiscarded list, and the priority of the current batch is 2, a new batch instance with priority 1 should be made with the products in the list, and all other parameters the same as the current batch except for the location. The parameters for the products should be reset, as they need to complete the sawing and labelling operations again, and new arrival events should be made and added to the FES. If the priority is not 2, the parameters of the products are reset and the products are added to the ProductsToBeScheduled list, and they will be batched according to Algorithm 4 later.

If there are still products left in the batch, then the next queue is determined based on the route, the corresponding service time is determined, and an arrival event for the batch is made and added to the FES. If the length of the queue is greater than the number of servers, the next batch can start processing. The next batch that should be processed is determined, the waiting time of this batch is registered, the next service time is taken, and a departure event for this batch is made and added to the FES. Lastly, a check is carried out to see whether a blockage in one of the preceding operations was caused by this machine, and products are waiting for this block to be over. If this is the case, the status of the preceding operation is changed, and the first batch that is waiting is picked. The waiting time of this batch is registered, the next service time is determined, and a departure event is made and scheduled to the FES.

7.1.4 External operation

Algorithm 1 holds for all operations except for the external operation. Once a week, on Tuesdays, the products are sent out to undergo the external operation. The next week on Thursday, they return to Berkvens, where they can continue their flow through the LDF. Algorithm 8 in Appendix 5 describes the process for dealing with the external operation. If the next queue is the external operation queue, the batch is appended to the *queueExternal* list. Every time a new day arrives, the day of the week is checked, and if this is Tuesday, the batches in the *queueExternal* list are sent out. A departure event is made for 7 days later and adds to the FES, and the *queueExternal list* is emptied.

Algorithm 1 Batch flow: Departure Event

```
if event.type == Departure AND event.location ≥ 4 then
    if machine is blocked, i.e. statusOperations[event.location] > 0 then
        Add batch to the list of products that are waiting for the block to be over
    else if Machine is closed with regard to its opening hours then
        Register the waiting time from this moment until the machine is operating again
        Make a departure event for the batch for the time the machine opens plus the service time
        Add the event to the FES
    else if Machine has a breakdown then
        Register the waiting time from this moment until the machine is operating again
        Make a departure event for the batch for the time the machine opens plus the service time
        Add the event to the FES
    else
        Remove the batch from the queue
        for Products in batch do
            Determine if product needs to be discarded or repaired
            if Product needs to be repaired then
                Add product to ToBeRepaired list
            end if
            if Product needs to be discarded then
                Add product to ToBeDiscarded list
            end if
        end for
        if Length ToBeRepaired > 0 then
            if batch.priority == 2 then
                Make a new batch instance with priority 1, the products in the ToBeRepaired list,
                  and all other parameters the same as the current batch
            else
                Make a new batch instance with the products in the ToBeRepaired list,
                  and all other parameters the same as the current batch
            end if
            Make an arrival event for the batch for 8 hours later
            Add the event to the FES
        end if
        if Length ToBeDiscarded > 0 then
            if batch.priority == 2 then
                Make a new batch instance with priority 1, the products in the ToBeDiscarded list,
                  and all other parameters the same as the current batch
                Reset all parameters for the products
                Make arrival events for all sawing and labelling activities that the product starts at
                Add arrival events to the FES
            else
                Reset all parameters for the products
                Add products to the ProductsToBeScheduled list
            end if
        end if
```

```
43:         if Length batch.products > 0 then
44:             Determine the next queue based on the route
45:             Determine the next service time based on the route
46:             Make an arrival event for the batch for the current time plus the flowtime at the
                    next queue with the next service time
47:             Add the event to the FES
48:         end if
49:         if Length queue ≥ number of servers at current location then
50:             Determine next batch in queue
51:             Register the waiting time
52:             Determine the next service time of this batch
53:             Make a departure event for the batch for the current time plus the service time
54:             Add the event to the FES
55:         end if
56:         if Preceding machine is blocked due to this operation AND batches are waiting for the
             block to be over then
57:             Change status of block for the operation regarded
58:             Pick the first batch that is waiting in the queue
59:             Remove the batch from the queue
60:             Register the waiting time
61:             Determine the next service time of this batch
62:             Make a departure event for the batch for the current time plus the service time
63:             Add the event to the FES
64:         end if
65:     end if
66: end if
```

7.1.5 Buffers and blockages

In the LDF there are buffers where the batches are stored in stacks before they can be processed on the next machine. These buffers are in place to ensure smooth flow between processes, such that operations can always continue processing. There is a possibility that a buffer is full, which causes blockages in the process, as operations that fill the full buffers cannot stash the batch anywhere. In the simulation, this possibility on full buffers and blockages is incorporated in the algorithm: in Algorithm 7 lines 4 till 7 the buffers are updated with every arrival and a check is carried out whether the buffers are full or not, and if this is the case, the preceding operation is blocked. In Algorithm 1 lines 2 and 3 a check is carried out to see if the operation is blocked due to a full buffer in the successive operation, and if this is the case the batch considered is appended to a separate list of batches that are waiting for the block to be over. In line 50, when a new departure event is made, first this list is checked and only if this list is empty, the next product in the normal queue is picked again.

7.1.6 Opening hours and rework and repair

As described in Section 4.6.6, some operations operate 16 hours per day, while others operate 24 hours per day. This is included in the simulation model as well by adding a list of opening hours for every operation. In Algorithm 1 this is shown in line 4 until 7, where a check is carried out to see if the machine is closed or open with regard to its opening hours. If the machine is closed, the additional waiting time for the batch is registered and a new departure event for the batch is made, but then for the time when the machine opens again plus the service time of the batch. As no other new departure events are made for this operation, all batches in the queue are waiting for this one batch to depart, which only happens when the machine is operating again.

The probabilities on repair and rework has been investigated in Section 4.6.1. The chance on rework and repair is already included in the main simulation, lines 14 until 42. When products need to be repaired, the need to visit the same operation again after eight hours, which is the average time it takes to repair a broken product. When products need to be discarded, they can either wait for the new product arrivals and be batched together with those, or they can start processing immediately by forming their own batch. This depends on the priority level of the batch and the remaining days until the due date.

7.1.7 Machine breakdowns

In Section 4.4 the breakdowns of the operations are investigated, which should be included in the simulation model as well. Algorithm 9 in Appendix 5 shows the procedure for determining breakdowns for every operation. First, two arrays with zeros are made, one to store the start times of the breakdown and one to store the end times of the breakdown. Then a random number between 0 and 1 is generated for every day in the length of the simulation. If this number is smaller than the probability of a day with a breakdown, there is a breakdown on that day. The length of this breakdown should be determined next, which is done by generating a random number with the exponential distribution and the parameters investigated for every operation, stated in Table 6. The start time of the breakdown is determined by generating a random number between the opening time and the end time minus the duration of the breakdown. This start time replaces the zero-value in the list for start times, and the end time replaces the zero-value in the list for end times. These lists are used when checking if there is a breakdown in lines 8 until 11 in Algorithm 1.

7.2 Model verification and validation

Simulation models should be verified and validated before conclusions can be drawn based on the output of the model. "Model verification is defined as ensuring that the computer program of the computerized model and its implementation are correct" ([Sargent, 2011], page 183), thus ensuring that the model works as intended. Model validation is concerned with the accuracy of the output of the model [Schlesinger, 1979], thus checking if the results of the simulation model are a correct representation of the reality. When the model is verified and validated with the current situation, the model can be used to test out different scenarios and improvements.

The model is verified by running a debugger, running the model for a small set of arrivals and small number of days, and running the model for 100 days and storing the results in an Excel file. The debugger was unable to identify problems, and the model ran smoothly without bumping into error that terminated the process. For the small run, a command was added to print all the events such that the changes in states can be followed manually. There were no unexpected events identified in this small run: the model behaved as expected, the products and batches flowed according to their routes, and all the checks and exceptions are working as intended. The results for the big run that were stored in the Excel file included the following parameters for every product: the product number, batch number, priority level, arrival time, due time, departure time, makespan, on time, service time, and waiting time. A part of the output file is stored in Table 34 in Appendix 6. Again, no unexpected events were identified. The reported average makespan, average waiting times, and service level were equal to the one calculated manually based on the results. The model thus worked as expected, thereby ensuring the model verification.

The model is validated by checking the results of the 100-day run done for the model verification. The average batch size and stack height is one of the output parameters, and are reported as 60.87 products and 21.39 products respectively, which is in line with the expectations from company experts. The average service level based on this 100-day run is 0.97, which indicates that some orders might be too late, but most of the orders are on time, which is in line with what the data suggests. Overall, the results seem reliable and close to the expected results based on the reality, which ensures the validity of the model.

7.3 Warm-up period, simulation duration, and number of runs

The simulation starts from scrap, meaning that there are no products present in the simulated factory. When there are no products in any of the operations, the waiting time and makespan for operations will be low, as products can be processed immediately. To ensure the results generated by the simulation model are close to the actual performance of the LDF, a warm-up period should be used, to ensure that the system is sufficiently filled with products. Furthermore, to ensure the system actually fills up, the number of arrivals in the first 10 days is manually increased by 1500 products per day. This helps the system to get to a steady state more quickly. The reported results are only based on the performance of the system after the warm-up period is over, where the system is in a stable and representative state.

To check whether the system has reached a stable state, the output of the 100-day run used to ensure the model validation and verification has been analyzed in more depth. Specifically, the daily arrivals, departures, and number of products in the system has been investigated, as well as the daily and overall service level, and the time the product spent in the system. The results are visualized in Figure 24, 25, and 26 respectively. From Figure 25 it can be seen that the service level of the products that arrived in the system fluctuates until day 20, and after that it remains stable for the rest of the simulation time. Furthermore, from Figure 24 it can be seen that the daily number of arrivals and departures, and the number of products in the system seems more stable from day 20 on. Lastly, from Figure 26 it becomes clear that also the time products spent in the system is stable after day 20. Therefore, the warm-up period is set to 20 days, which means that the results are calculated from day 20 onwards.

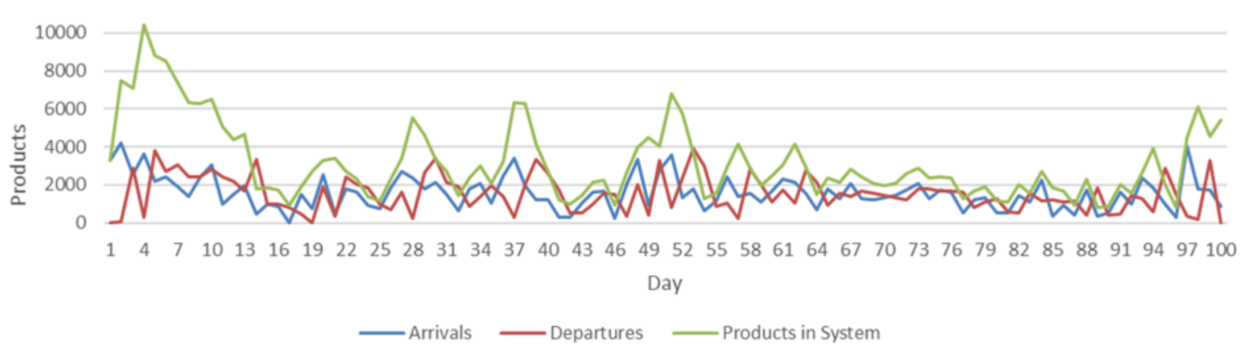


Figure 24: Warm-up: Daily arrivals, departures, and products in system

When the system is deemed stable, the output of the simulation model from that moment on represents the actual performance of Berkvens' LDF. The simulation model should be run multiple times to ensure the results obtained are representative of the reality. When only one run is done, the

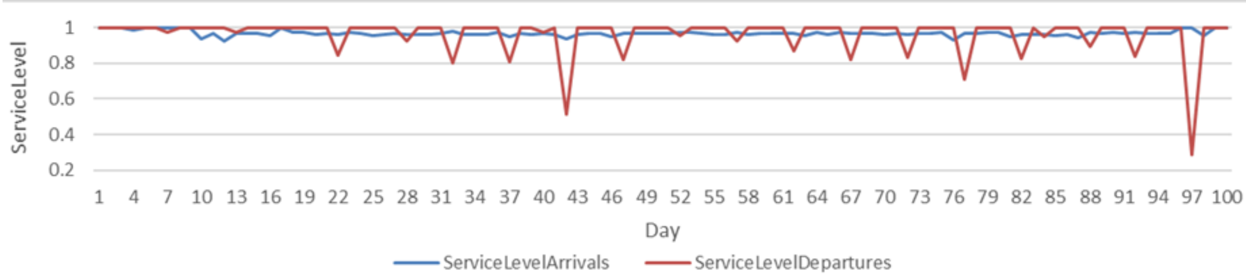


Figure 25: Warm-up: Service Level of daily arrivals and departures

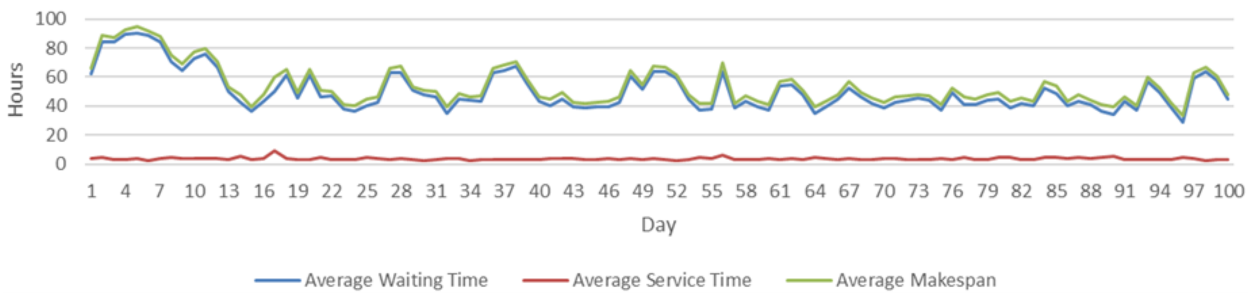


Figure 26: Warm-up: Average time spent in system

results of this run might be skewed by a large disruption in one of the processes, or an exceptionally high number of arrivals. When the simulation is run over 30 time periods, the central limit theorem can be used to construct a confidence interval around the results obtained by the simulation model with the Normal Distribution. The central limit theorem states that if the sample taken from a population is sufficiently large, the sample means will follow a normal distribution. The minimal number of observations for the central limit theorem to apply is 30 [Mann, 2016] [Montgomery and Runger, 2018], therefore the simulation should run 30 times. To save time, the system is initialized only once and the system is run for a lengthy period of time. This period of time is divided into smaller blocks, which all represent one run. The results are obtained from each block separately, thus only the results of the products that left the system during the block observed are used to obtain the mean results of that block. Thus, the simulation should run for a warm-up period of 20 days plus 30 blocks of the same length to ensure the obtained results are representative of the system.

The length of each run of the simulation model should be determined next. One requirement for the central limit theorem to apply is ensuring the observed means are independent. However, running for a long time will comprise the efficiency of the system, resulting in long running times. Therefore, a trade-off needs to be made, to ensure the runs are independent, while maintaining a reasonable computation time. The average time a product spends in the system is 74 hours, based on early output. Thus when one simulation run lasts 10 days, almost all products will have left the system, which ensures independent runs. Therefore, the length of one block in the simulation run is equal to 10 days. This should ensure the independency of the blocks and thus obtain reliable results about the system.

7.4 Model execution for current situation LDF

The model is executed for the indicated simulation length, i.e. 20 days warm-up plus 30 blocks of 10 days makes 320 days of simulation. This simulation has the parameters of the current situation of the LDF, where the sequencing depends on the priority of the batches. In Table 18 the most important results are presented and in Table 19 the results on the service level for MTO- and MTS orders are given. In the first table, all time variables are in hours. In Appendix 7 some additional simulation results are given. A 95% confidence is constructed around all the means with the normal distribution, which can be used because of the central limit theorem. Equation 22 is used to construct the confidence interval (CI), where $\bar{x}$ is the average, s is the sample standard deviation, n is the number of observations, and z is the confidence level value with the standard normal distribution, which is 1.96.

A couple of things are interesting based on these results. Firstly, the average service level is 0.9081 and the confidence interval has a lower bound of 0.8591 and an upper bound of 0.9571. This means that it is very unlikely that the KPI of a 98% internal service level will be reached. Furthermore, the service level for the MTO orders is significantly lower than the service level for MTS orders, which is undesirable as a high service level for MTO orders results in fewer penalties for lateness. The reported service level for the MTO orders is 0.881, which is very close to the actual service level of MTO orders measured in Section 2.2, which was 0.885, further validating the simulation model. Secondly, the average makespan is only 74.40 hours, which is a little over 3 days. This means that many products flow quite quickly through the system, and indicates that many of the products that are late have a priority of either 1 or 2. Lastly, the waiting time contributes to almost 95% of the total makespan, which means that products spent most of their time waiting instead of actual production.

Parameter	**Average**	**St. Dev.**	**Confidence Interval**	
			Lower bound	**Upper bound**
Service Level	0.9081	0.1346	0.8591	0.9571
Makespan	74.40	53.98	54.75	94.04
Waiting Time	70.65	53.97	51.01	90.30
Service Time	3.74	1.13	3.33	4.15

Table 18: Results current sequencing method LDF

Parameter	**Average**	**St. Dev.**	**Confidence Interval**	
			Lower bound	**Upper bound**
Overall	0.9081	0.1346	0.8591	0.9571
MTO	0.8806	0.3242	0.7646	0.9966
MTS	0.9940	0.0771	0.9664	1

Table 19: Results MTO and MTS current sequencing method LDF

$$\mathbf{CI} = \bar{x} \pm z \frac{s}{\sqrt{n}} \tag{22}$$

In the next chapter, three (meta)heuristics are described, which may increase the performance of the LDF. These three (meta)heuristics will be tested with the simulation model described in this chapter, and their results will be described in Chapter 9.

8 Heuristics and metaheuristics for the sequencing problem

In this chapter three (meta)heuristics that will be tested are described. This is done to help answer the second part of SRQ 4: "...what improvements can be made in the sequencing with the objective to minimize the tardiness and makespan of orders?". In Section 3.3 the results of the literature review [Jeuken, 2022] are briefly shared. In short, this section indicated that the best method for solving a HFS problem, such as encountered in this research, is using heuristics or metaheuristics. The most frequently used heuristics in the reviewed articles were dispatching rules, and the most frequently used metaheuristics were Simulated Annealing (SA), Tabu Search (TS), and Genetic Algorithm(GA), which are described briefly in Section 3.3. Dispatching rules are easy to implement in a real-world system, and therefore two of the proposed improvement methods are forms of dispatching rules, namely the Earliest Due Date (EDD) dispatching rule, and the RR dispatching rule. The EDD-rule is fairly straightforward and easy to understand, while the RR-rule is more complicated and a more tailored dispatching rule. To compare the performance of those dispatching rules with a more sophisticated method, SA is chosen as the third improvement method. Only one metaheuristic is chosen, as metaheuristics take in general longer to tune and compute. Furthermore, dispatching rules are easier to implement in a complex system, such as the LDF of Berkvens. SA is chosen over GA and TS as multiple articles reviewed in the literature review [Jeuken, 2022] indicated that this method performs well in systems comparable to the situation at hand. In the remainder of this chapter, the three methods are described in detail.

8.1 Earliest Due Date

Dispatching rules are widely used in practice, due to their ease of use and intuitive nature [Aurich et al., 2016]. Furthermore, they are very suitable to deal with complex, dynamic, and unpredictable environments and are thus very popular in practice [Ruiz and Vázquez-Rodríguez, 2010]. Dispatching rules basically determine the order in which all jobs considered should be processed [Hopp and Spearman, 2011a]. In the literature research done as preparation for this thesis [Jeuken, 2022], various dispatching rules were encountered, including amongst others Shortest Processing Time (SPT) rule, Earliest Due Date (EDD), and Minimum Slack Time (MST), which are popular rules in practice. The main objective of this research is to increase the service level of the LDF of Berkvens. The EDD rule prioritizes the orders with the earliest due date and is often used when the objective is to minimize the maximum lateness [Nahmias and Olsen, 2015], which is the main objective of this research. This priority rule probably performs well in the environment encountered in this research, as the goal is to minimize the tardiness and the EDD rule tries to prioritize orders such that no order is tardy: the product or batch that has its due date closest to the current time, receives the highest priority and is processed first.

8.2 The RR rule

Besides the well-known dispatching rules mentioned in the previous section, many other dispatching rules have been developed over the years, and the choice of one dispatching rule over another depends on the objective of the flow shop. In this research, the primary objective is to minimize the number of tardy jobs, with the secondary objective of minimizing the makespan. According to [Xiong et al., 2017], who performed a simulation based study on the performance of several dispatching rules, "RR shows dynamic and global characteristics, and demonstrates to have superior performance in flow time and tardiness related measures" ([Xiong et al., 2017], p. 17). The RR rule is developed in the early 1990s by Raghu & Rajendran [Raghu and Rajendran, 1993], and considers the processing time, slack, and waiting time for the next operation. The RR rule seeks to minimize the mean flowtime and mean tardiness of jobs. The RR rule outperforms several other dispatching rules when the utilization rate of the operations is smaller or equal to 85%. The utilization rates have been calculated in Section 4.6.6, and the utilization rates of the drilling operation, the CNC operation, and the SPEC operation is a little higher than 85%, namely 88.7%, 87.7%, and 86.2% respectively, which might influence the performance of the RR-rule. Furthermore, the RR rule has the best performance in terms of the mean tardiness as well, compared to many other dispatching rules [Rajendran and Holthaus, 1999], [Xiong et al., 2017], [Raghu and Rajendran, 1993]. Thus, the RR-rule will probably have a high performance in the manufacturing environment encountered in the research, and will thus be tested by means of simulation.

The RR rule thus considers the processing time, slack, and waiting time for the next operation. The priority index of a job, Z_i, is given according to Equation 23 [Raghu and Rajendran, 1993] [Xiong et al., 2017]. Here, η is the utilization level of the machine on which job i will be loaded, RPT_i is the sum of process times of uncompleted operations including the current operation, W_{nxt} indicates the probable waiting time of job i at the machine of its next operation considering its priority, s_i is the slack of job i, and t_{ij} is the process time of operation j of job i.

$$Z_i = \frac{s_i \cdot e^{-\eta} \cdot t_{ij}}{RPT_i} + e^{\eta} \cdot t_{ij} + W_{nxt} \tag{23}$$

8.3 Simulated Annealing

Simulated Annealing (SA) is introduced in 1983 by Kirkpatrick et al. [Kirkpatrick et al., 1983]. This metaheuristic is based on annealing, a process of cooling molten metal to increase its strength, from which it also derives its name [Luke, 2013]. SA is an iterative, stochastic, neighborhood-based search method and has been widely applied to solve combinatorial optimization problem [Behnamian et al., 2010], such as the Hybrid Flowshop Scheduling (HFS) problem. SA is implemented in various papers, such as [Aurich et al., 2016], [Sun et al., 2011], and [Baxendale et al., 2021]. This metaheuristic is more complex than the dispatching rules explained in the previous sections and has more parameters and variables, which are explained and determined below.

The algorithmic framework of SA is described in Algorithm 2. An initial solution S is taken and its fitness is determined with the fitness function $f(S)$. While the termination conditions are not met, a neighbor solution S" is generated and the fitness of this new solution is determined with $f(S")$. If the fitness of this neighbor solution is better than the original solution, S" becomes S. To decrease the chance of getting stuck in a local optimum, the non-optimal solution is selected with probability $e^{\frac{f(S")-f(S)}{t}}$, where t is the temperature. With every iteration the temperature is decreased, such that the probability of accepting an inferior solution decreases with the number of iterations. The main idea of SA is to initially set t to a high number, such that the whole solution space is searched before settling on an optimal solution.

In short, SA considers an initial solution and tests its fitness, generates neighbor solutions and tests their fitness, and selects the solution with the best fitness as optimal. In this research, SA will be used to generate an optimal sequence for products and/or batches at the various stages. There are thus multiple local solutions, i.e. the sequence for products or batches at each stage, which together form the overall solution. The SA algorithm will be executed for every stage individually instead of the whole LDF, to account for the speed and complexity encountered in real life. This complexity is due to the repair and rework probabilities investigated in Section 4.6.1, which causes changes in the set of jobs that need to be processed. Basically, every time a product or batch moves to a new

Algorithm 2 Simulated Annealing, *source: [Luke, 2013]*

```
t ← temperature
while termination condition not met do
    S” ← Pick neighbor at random and determine fitness
    if f(S”) ≤ f(S) or a random number chosen from 0 to 1 < e^((f(S”)−f(S))/t) then
        S ← S”
    end if
    decrease t
end while
best ← S                                         ▷ Output: best
```

operation, a new optimal solution may arise. Changes in locations take place all the time in the LDF, therefore generating a new sequence for the whole LDF is illogical, and instead a new local optimal sequence for the operation that has a new arrival should be generated. This need for a local optimal sequence brings a new challenge: determining the fitness of a local sequence, which will be explained later on.

From the algorithmic framework of SA, it becomes clear that various parameters should be determined. First of all, the initial solution should be determined. The fitness of the initial solution should be calculated with a fitness function. Neighbors of a given solution should be generated and selected. The initial temperature and the cooling scheme should be determined. Lastly, the maximum number of iterations should be determined. The determination of all these parameters is described in more detail in the following subsections.

8.3.1 Initial solution and neighbor generation

The initial solution is created by implementing the EDD dispatching rule, explained in Section 8.1. Neighbor selection is a critical part of the simulated annealing algorithm, as this determines which possible solutions are being evaluated. There are three common methods for generating neighbor solutions in simulated annealing [Glover and Laguna, 1998] [Dowsland and Thompson, 2012] [Bayram and Şahin, 2013]:

- Random perturbation: In this method, the current solution is randomly modified by making small changes to one or more variables. One of the main advantages is that the entire search space can be explored and local optima are efficiently escaped. However, the random perturbation can be slow and inefficient, especially when the search space is large and complex. It often generates solutions that are far from optimal and thus more iterations are required to merge.
- Local search: In this method, the current solution is modified by making small changes that improve the objective function locally. Local search can lead to faster convergence and better results when the search space is smooth and there are many local optima. Furthermore, it is mostly effective when the initial solution is already close to optimal. However, local search may not be able to explore the entire search space and can thus get stuck easier in local optima.
- Problem-specific methods: Besides the random perturbation and local search methods, also problem-specific methods can be implemented.

For this problem, the local search neighbor generation will be used. This is done because local search can lead to faster convergence and is typically very effective in finding high-quality solutions when the initial solution is already close to the optimal solution. As the EDD rule will be used to generate the initial solution, the initial solution will probably be close to optimal already. The local search algorithm is hence an effective means to finding good neighbor solutions.

8.3.2 Fitness determination

The next parameter that should be determined for the simulated annealing metaheuristic is the determination of the fitness. The fitness of a sequence consists of two factors, namely the tardiness and the makespan of that sequence. As stated before, the final solution is comprised of multiple local solutions, which are generated at each stage of the production process. This means that the tardiness and makespan should be determined on local level as well. To do this, the due date and due time of a product is determined for every operation. This is done by looking at the route of the product, the average waiting times at every operation in a products' route, and the time between release and due time of a product. Based on these factors, the due time and date per operation can be determined for every operation in the route of a product. The fitness function is the sum of the tardiness of all products plus the sum of the processing times of each product and the processing times of all products that precede a product in the sequence. The fitness function $f(S)$ is given in Equation 24, where P_j is the processing time of job j and D_j is the due time of the product for the considered operation. S is the sequence, for instance S = [1, 3, 4, 2, 6, 5], where s is each element of S, ordered from left to right.

$$f(S) = \sum_{i=0}^{S} [P_{j-i} - D_{j-i}]^{+} \cdot 0.9 + \sum_{s}^{S} \sum_{i=0}^{s} P_{j-i} \cdot 0.1 \tag{24}$$

8.3.3 Initial temperature and cooling scheme

The initial temperature and the cooling scheme are two parameters that are highly important for the performance of the simulated annealing metaheuristic. The initial temperature determines the probability of accepting a worse solution in the early stages of the search. Setting the initial temperature too high or too low may result in poor performance or slow convergence. A common method for determining the initial temperature is the empirical method. In this method the simulated annealing algorithm is run a couple of times with different initial temperatures, where the performance of each iteration is tracked. Based on the performance of each iteration, the initial temperature is determined.

The cooling scheme determines how the temperature decreases over time. It affects the efficiency and effectiveness of the algorithm over time. There are some common cooling schemes that are often used in simulated annealing [Aarts and Korst, 1989] [Kirkpatrick et al., 1983] [Velleda Gonzales et al., 2015]:

- Linear cooling: in this scheme the temperature decreases linearly over time. The temperature at each iteration is a fraction of the initial temperature and the fraction decreases linearly over time. Advantages of this cooling scheme are the ease of implementation and the ease to control and tune. Disadvantages are that the cooling rate may require quite some fine tuning, as it may converge too quickly or too slowly;
- Exponential cooling: in this scheme the temperature decreases exponentially over time. The temperature at each iteration is a function of the initial temperature and a cooling rate. The

main advantage of this cooling scheme is the adaptability, which provides a good balance between exploration and exploitation. Disadvantages are the sensitivity to the initial temperature, and the need for a warm-up period. Furthermore, this method also requires fine tuning to prevent from converging too slowly or prematurely;

- Boltzmann cooling: in this scheme the temperature is a function of the current energy level of the system. The temperature is decreased based on the Boltzmann distribution. The main advantage of this cooling scheme is the flexibility of the approach, which exceeds the flexibility of the linear and exponential cooling schemes as it can adapt to the characteristics of the problem considered. Disadvantages are that the Boltzmann cooling scheme is generally computationally expensive, as it requires calculating the acceptance probability of each new solution. Furthermore, it may converge too slowly in some problems or converge prematurely in others.

Based on the advantages and disadvantages of the cooling schemes mentioned above, the linear cooling scheme seems most fitting in this case. Firstly, this cooling scheme is fairly straightforward and easy to implement with a reasonable computation time, which is needed due to the size of the network. Secondly, this cooling scheme will probably perform well, as the initial solution probably has a high performance, so even if the algorithm converges prematurely, the solution should still be near optimal.

8.3.4 Maximum number of iterations and final temperature

The maximum number of iterations in simulated annealing is usually determined based on the available computing resources and the expected convergence rate of the algorithm for the specific problem being solved. In practice, the maximum number of iterations is often determined through trial and error, by experimenting with different values and evaluating the algorithm's performance on the specific problem being solved. It is also common to set a relatively large maximum number of iterations and then stop the algorithm when a satisfactory solution is found, or when the algorithm reaches a plateau in performance.

In the previous section, the choice for a linear cooling scheme has been made, thus the temperature decreases linearly over time. Therefore, either the number of iterations or the final temperature should be determined, as they can be deviated from each other due to their relation, which is as follows: cooling rate $= \frac{\text{Initial temp} - \text{final temp}}{\text{nr. Iterations}}$. Thus, either the cooling rate and final temperature should be determined, or the maximum number of iterations and the final temperature.

As stated before, the network is complex and many arrivals and departures occur in a short time frame. To ensure the computation time needed for the simulation is limited, the metaheuristic is not executed every time a new product or batch arrives, but only when 10 products or batches have arrived. This will ensure the metaheuristic can be tested, but limits the time needed to complete the simulation.

8.4 Chapter conclusion

In this chapter, the second part of SRQ4 stood central, which is "... what improvements can be made in the sequencing with the objective to minimize the tardiness and makespan of orders?". Three methods that could lead to improvements in the sequencing have been introduced and described in this chapter. These three will be compared with each other and with the current performance of the LDF of Berkvens with the simulation model developed in Chapter 7. The results of these simulations will be described in the next chapter.

9 Results new sequencing methods

The previous chapter introduced three (meta)heuristics that could lead to improvements in the sequencing of the orders in the LDF of Berkvens. In this chapter the results on the performance of these three are compared and the best one is chosen. To be able to compare the results of the three methods fairly, they should be tested over the same data. Therefore, the arrival data, data about breakdowns, and data about repair and rework should be simulated beforehand and used as input in the main simulation model. Only when those factors are the same for all simulations, the results of the various methods can be compared and valid conclusions can be drawn from the simulations. This data is created for 320 days, 20 days warm-up plus 30 blocks of 10 days.

9.1 EDD rule

The most important results of the simulation output of the EDD priority rule are given in Table 20. In Appendix 8 some additional results for every operation are given on the waiting time, queue length, and periods of time that a machine is running uninterrupted.

From Table 20 it can be seen that the average service level increases to 94% when the EDD rule is implemented. A 95% confidence interval (CI) has been constructed around the average service level, which spans from almost 92% to 95.6%. This implies that the average service level will definitely increase compared to the current average service level of 90.8%. The KPI of a service level of 98% will however not be achieved with this priority rule, as the upper bound of the 95% CI is 95.63%.

Besides the service level, also the average makespan is stated in Table 20, which is 69.32 hours. Over 95% of this time is comprised of waiting time, and only roughly 5% of this time is used for the actual processing of products. This makespan is a slight improvement compared to the current makespan, which is on average 74.4 hours. The 95% CI that is constructed around the mean has a width of approximately 10 hours, and the upper bound of 71.3 hours indicates that it in general, it should be possible to produce almost all product within 4 days (96 hours), thus the products with the highest priority should normally be completed on time.

Parameter	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Service Level	0.9381	0.0508	0.9199	0.9563
Makespan	69.32	14.57	64.11	74.54
Waiting Time	66.05	14.68	60.80	71.31
Service Time	3.34	0.33	3.23	3.47

Table 20: Results EDD-rule

Parameter	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Overall	0.9381	0.0508	0.9199	0.9563
MTO	0.9024	0.2968	0.7962	1
MTS	1	0	1	1

Table 21: Results service level MTO and MTS EDD-rule

9.2 RR rule

The most important results of the simulation output of the EDD priority rule are given in Table 22. In Appendix 9 some additional results for every operation are given on the waiting time, queue length, and periods of time that a machine is running uninterrupted.

From Table 22 it can be seen that the average service level based on this heuristic is 84.41%, with a confidence interval of 78.25% until 90.57%. Compared to the service level of the current situation, which is 90.81% on average, it seems that this method will probably not bring an improvement to the service level of the LDF. Moreover, if this heuristic would be implemented, there would probably be a decrease in performance. This 'bad' performance could be the result of the utilization rates being above 85% for some operations, as the literature indicated that the RR-rule performs well with utilization rates of below 85% [Xiong et al., 2017].

The average makespan reported in Table 22 is almost 105.3 hours, which is significantly higher than the current makespan of the LDF. Again, the majority of this time is made up by the waiting time, over 96%. When looking at the standard deviation of the makespan, it becomes clear that this method creates high variability in the output. This is undesirable, as the prediction of the makespan becomes harder when a process is more variable. Furthermore, higher variability in makespan means that disruptions in the process cause more impact.

Parameter	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Service Level	0.8441	0.1722	0.7825	0.9057
Makespan	105.29	59.72	83.92	126.66
Waiting Time	101.45	59.57	80.13	122.77
Service Time	3.34	0.33	3.23	3.47

Table 22: Results RR-rule

Parameter	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Overall	0.8441	0.1722	0.7825	0.9057
MTO	0.7582	0.4281	0.6050	0.9115
MTS	0.9970	0.0549	1	1

Table 23: Results service level MTO and MTS RR-rule

9.3 EDD rule per operation

As described in Section 8.3, the Simulated Annealing metaheuristic will be executed for every stage individually. To accommodate this, a due date for every product/batch is determined on every operation it needs to flow through. This due date is mainly used as input for the fitness function needed for the Simulated Annealing algorithm. However, the EDD rule, as explained in Section 8.1, can be used with this new due date per operation as well. Thus, the simulation model with 30 batches, 10 days per batch, and 20 days warm-up time has been executed with the EDD rule with a due date per operation.

The due date per operation is determined by taking the average waiting times per operation based on the EDD rule, which are stated in Table 38 in Appendix 8. First a due date for the sawing and labelling activities is determined for each product. This is done by adding up the service time and the waiting time for every operation that the product needs to be processed by. The deadline for each operation is then determined by adding this time to the arrival time of the product. The overall deadline for the sawing and labelling operation is determined by taking the maximum of the service times plus waiting time for each operation and multiplying this with a factor of 0.1 to account for some disruptions.

This deadline for the sawing and labelling activities is then used as input for the other operations, where products are processed in batches. Here, the deadlines are determined slightly different. First, the total waiting times of the operations the batch visits and the service times are summed. Then the remaining time is calculated by subtracting the total waiting time, the total service time, and the deadline for the sawing and labelling activities from the due time of the batch. Note that this remaining time may be negative. The remaining time is then divided over the operations that the batch visits, where the proportional waiting time per operation is taken into account. For instance, if drilling operation makes up 20% of the total time the batch still needs to spent in the system, 20% of the remaining time will be assigned to the drilling operation. This results in a due date for all operations the batch needs to visit.

The results of the EDD per operation heuristic are shown in Table 24 and again some additional results on the individual operations are reported, see Appendix 10. From Table 24 it can be seen that the average service level is as high as 0.9459, which is well above the current service level. When looking at the 95% confidence interval, it becomes clear that it is very unlikely that the KPI of a 98% service level will be reached with this heuristic. When looking at the makespan, there is quite some improvement with respect to the current average makespan of 74.4 hours, which is decreased by 7 hours. The computation time of this heuristic is ...

Parameter	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Service Level	0.9459	0.0287	0.9356	0.9561
Makespan	67.42	12.06	63.10	71.73
Waiting Time	64.86	11.91	60.59	69.12
Service Time	3.34	0.33	3.23	3.47

Table 24: Results EDD rule per operation

Parameter	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Overall	0.9459	0.0287	0.9356	0.9561
MTO	0.9161	0.2773	0.8169	1
MTS	1	0	1	1

Table 25: Results service level MTO and MTS EDD rule per operation

9.4 Simulated Annealing

Several parameters should be determined before the Simulated Annealing metaheuristic can be tested with the simulation model, namely the initial and final temperature, the cooling rate, and the maximum number of iterations. First, the decision process for these parameters is described. Then, the results of the simulation with the SA metaheuristics are given.

9.4.1 Parameter determination

The decision on the parameter values for the SA is basically a trade-off between the computation time and the quality of the solution. When a large part of the search space should be searched, the solution that the algorithm comes up with will be good in general, but the computation time will be long. However, when only a small part of the search space is explored resulting in a shorter computation time, the solution might not be optimal due to the algorithm getting stuck in a local optimum. The choices of the parameters should therefore provide a good balance such that a good result is obtained within a reasonable computation time.

The initial temperature is typically determined first, as it is the starting point of the algorithm and can have a significant impact on its behaviour. The temperature is an important parameter, as the probability of accepting a neighbour solution as current solution is determined by the temperature and the results of the fitness function. Thus, when determining the initial temperature, the fitness function should be taken into account as well. The approximate value that the fitness function takes depends on the operation the product or batch is in and the number of products that are in the queue for this operation. Based on some initial runs of the simulation model with the SA metaheuristic, the values for the fitness function can be in quite a big range, from 0.1 to over 2500. It seems however, that the improvements obtained by neighbour solutions are in general quite small, less than 5% of the value of the fitness function of the current solution. This indicates that a high initial temperature is unnecessary: as the probability on large improvements is small, having a large initial temperature has no additional value, as the probability of accepting the neighbor solution is almost zero then. To determine the best initial temperature, cooling rate, and final temperature, several configurations were tested. These configurations are shown in Table 26. The simulation model has been adapted slightly to be able to compare the results of these configurations: the number of daily arrivals has been fixed to 1700, and the standstills have not been taken into account. The simulation script is initialized with the EDD heuristic to allow for a faster warm-up period, and each configuration is simulated for 2 days.

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
In. temp.	10	9	8	7	6	5	4	3	2.4	1.92	1.54	1.23	0.98	0.79	0.63	0.5
Cooling r	0.475	0.428	0.38	0.333	0.285	0.238	0.19	0.143	0.114	0.091	0.073	0.058	0.047	0.037	0.03	0.024
Fin. temp.	0.5	0.45	0.4	0.35	0.3	0.25	0.2	0.15	0.12	0.1	0.07	0.06	0.05	0.04	0.03	0.025

Table 26: Configurations for testing parameters SA: test 1

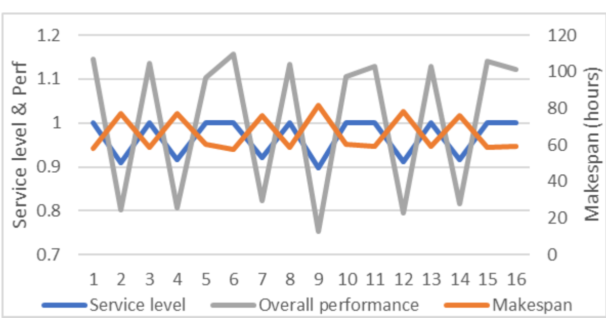


Figure 27: Parameters SA test 1: results performance

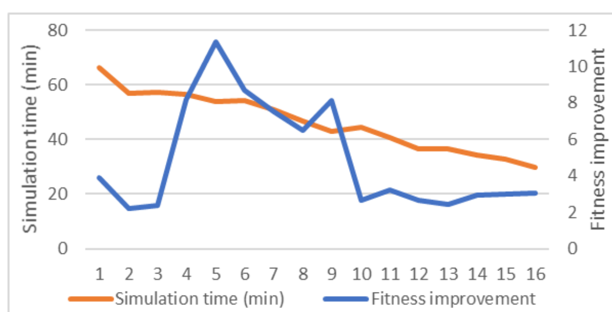


Figure 28: Parameters SA test 1: results computation time and fitness improvement

In Figure 27 and 28 the results on performance, computation time, and fitness improvement per configuration are given. When looking at the performance, it becomes clear that the changes in performance do not seem to follow a logical pattern. The blue line in Figure 27 represents the average service level, the orange line the average makespan, and the grey line a combination of these two parameters, which is determined by the formula: 1/((1−service level)∗(1−0.015)+makespan∗0.015). Here, the factor 0.015 represents the magnitude of the makespan compared with the service level. A clear relation between the service level and the makespan exists: when the service level is high, the average makespan is low and the other way around. When the correlation between those two is calculated, this results in an almost perfect negative correlation of -0.9965, which means that the higher the service level, the lower the makespan, and the other way around. Overall, based on the performance of the configurations with regard to the makespan and the service level, not one configuration has a clear advantage over another configuration.

When looking at the results of the computation time and the average improvement in fitness displayed in Figure 28, there clearly is a pattern in the computation time, which decreases when the initial temperature and the cooling rate decrease. Furthermore, the average improvement in the fitness of the runs is high in configurations 4 until 9, but low in the other configurations. However, when looking at the performance of these six configurations, it seems that the performance of these four configurations is similar to the performance of the other configurations. There is thus no logical explanation for the increase in average fitness improvement for these six runs. Overall, it seems no configuration has a clear advantage over any of the others based on all parameters except computation time, which might change when the cooling rate for the configurations is changed. Therefore, a second set of configurations is tested, which are shown in Table 27. The first three configurations from the first set are not included, as these do not perform better than any of the other configurations, but have a higher computation time, thus these parameters will not be chosen anyway.

Run	1	2	3	4	5	6	7	8	9	10	11	12	13
In temp.	7	6	5	4	3	2.4	1.92	1.54	1.23	0.98	0.79	0.63	0.5
Cooling r	0.665	0.57	0.475	0.38	0.285	0.228	0.182	0.147	0.117	0.093	0.075	0.06	0.0475
Fin. Temp.	0.35	0.3	0.25	0.2	0.15	0.12	0.1	0.07	0.06	0.05	0.04	0.03	0.025

Table 27: Configurations for testing parameters SA: test 2

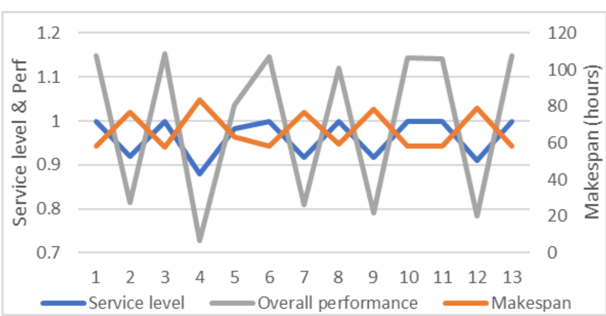


Figure 29: Parameters SA test 2: results performance

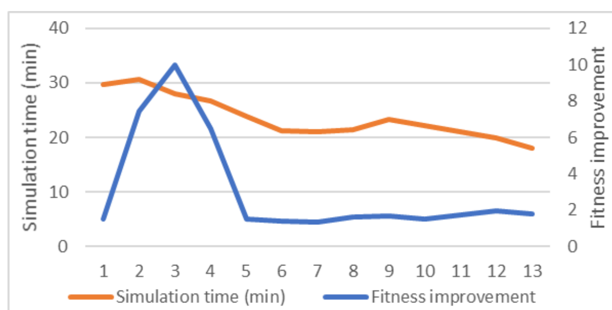


Figure 30: Parameters SA test 2: results computation time and fitness improvement

In Figure 29 and 30 the results on performance, computation time, and fitness improvement per configuration for the new configurations are given. From Figure 30 it can be seen that the fitness improvement is again higher for configurations 2, 3, and 4, which were configuration 5, 6, and 7 in the first test set. Specifically, for configuration 3 the fitness improvement is remarkably high, while the computation time seems to follow the normal decreasing pattern. In test set 1, this was configuration 6 which scored high in this test set as well. When looking at the performance of this configuration, it becomes clear that the performance is high in both test sets. Therefore, this configuration with a initial temperature of 5 and a final temperature of 0.25 seems the best configuration for the SA metaheuristic.

Now the initial and final temperatures are determined, the cooling rate should be determined next. As stated in Section 8.3.4, there is a relationship between the number of iterations and the cooling rate, so when the number of iterations needed is determined, the cooling rate can be derived from this. To determine these two parameters, various combinations of the number of iterations and the cooling rate have been tested, where the performance, computation time, and fitness improvement are measured. In Table 28 the combinations that are tested are given, and in Figure 31 and 32 the results of the tests are shown. From these figures it can be seen that there is a large decrease in simulation time when the number of iterations are decreased from 20 to 10, while the overall performance stays roughly the same. Furthermore, the simulation time keeps decreasing when there are fewer iterations, while there is a surge in performance when the number of iterations is 7, 5, and 4. Overall, the higher the number of iterations, the higher the performance, and as there are no large differences in computation time or performance, the number of iterations that will be used for the final simulation is determined to be 7.

Thus the parameters that will be used in the simulation of the SA metaheuristic are the following: an initial temperature of 5 and a final temperature of 0.25, maximum number of iterations is 7, and a cooling rate of 0.679. In the next section, the simulation results of the SA with this setup are described.

Nr. of iterations	20	10	9	8	7	6	5	4	3
Cooling rate	0.238	0.475	0.528	0.594	0.679	0.792	0.950	1.188	1.583

Table 28: Combinations number of iterations and cooling rate

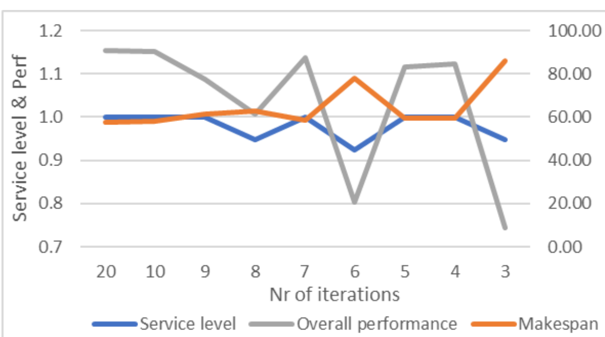


Figure 31: Cooling rate SA: results performance

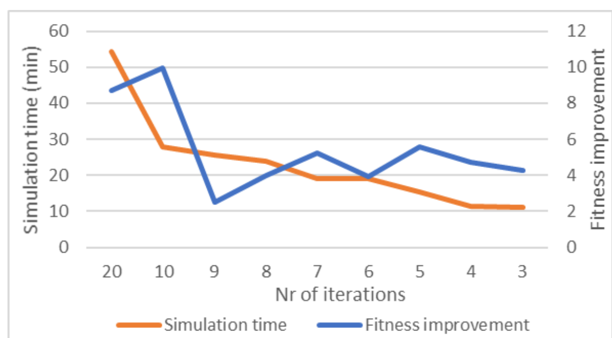


Figure 32: Cooling rate SA: results computation time and fitness improvement

9.4.2 Results Simulated Annealing

The results of the simulation of the SA metaheuristic are given in Table 29 and in Appendix 11 some additional simulation results are given. The average service level and average makespan are again improved compared to the current situation. However, also with this method, reaching the KPI of a 98% service level seems unlikely, as this value is excluded from the 95% confidence interval on the service level. The makespan is on average 67.35 hours and has an upper bound of 71.95

hours, which means that most products will have left the LDF within 3 days. Again, the waiting time makes up most of the makespan, approximately 95%.

Parameter	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Service Level	0.9470	0.0246	0.9382	0.9558
Makespan	67.35	12.84	62.76	71.95
Waiting Time	64.01	12.74	59.45	68.56
Service Time	3.34	0.33	3.23	3.47

Table 29: Results SA

Parameter	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Overall	0.9470	0.0246	0.9382	0.9558
MTO	0.9163	0.2684	0.8201	1
MTS	1	0	1	1

Table 30: Results service level MTO and MTS SA

9.5 Comparison (meta-)heuristics

Now the results of the four heuristics and metaheuristic are known, a comparison can be made between those four and the method that suggests the highest improvement can be identified. In Table 31 the most important results of each candidate improvement are summarized, and also the current performance of the LDF is given. In Figure 33 the performance of the (meta)heuristics in terms of the overall service level, the service level for MTO orders, and the makespan are visualized.

The RR-rule performed worst, which is a surprising result, as the literature indicated that this heuristic would outperform most other dispatching rules, including the EDD rule. This gap in performance could be explained by the nature of the LDF, which is a complex environment. The RR rule calculates priority based on several parameters, including the sum of the process times of uncompleted operations and the slack. The nature of this dispatching rule ensures that products will wait long in early operations if they have a long slack, which could result in being too late if the product needs to be repaired or discarded. This would be a logical explanation for the low performance of the RR-rule compared to the EDD-rule.

Based on the results shown in Figure 33 and Table 31, it seems that the SA metaheuristic has the best performance, closely followed by the EDD per operation rule. It is logical that the SA metaheuristic performs better than the EDD per operation rule, because the SA takes the sequence based on the EDD per operation rule as initial solution and improves this sequence. The improvement in performance is only small: the average service level improves with 0.001 and the average makespan with 0.07 hour. The confidence intervals are also similar, with only minor differences in the makespan confidence interval, which is slightly larger for the SA metaheuristic. As the performance of these two parameters is very similar, an additional parameter should be taken into account, which is the computation time. Overall, the SA metaheuristic took 42.35 hours to simulate, while the EDD per operation heuristic took only 7.51 hours. The EDD per operation rule thus gave similar results in a smaller time frame.

	Avg. Service level	Avg. Service level MTO	Avg Makespan	CI Service level		CI makespan	
Current	0.908	0.881	74.4	0.859	0.957	54.75	94.04
EDD	0.938	0.902	69.40	0.920	0.956	64.11	74.54
RR	0.844	0.758	105.29	0.782	0.906	83.92	126.66
EDD per op.	0.946	0.916	67.42	0.936	0.956	63.10	71.73
SA	0.947	0.916	67.35	0.938	0.956	62.76	71.95

Table 31: Comparison simulation results (meta)heuristics

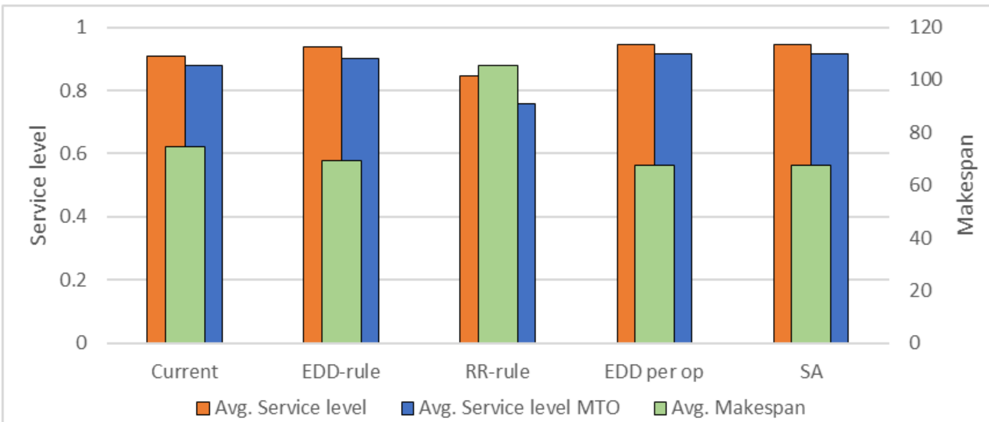


Figure 33: Comparison (meta)heuristics

9.6 Chapter conclusion

In this chapter, the heuristics and metaheuristic developed in the previous section are simulated and their performance is determined and compared. This is done to answer the second part of the fourth SRQ, which is: "... what improvements can be made in the sequencing with the objective to minimize the tardiness and makespan of orders?". Based on the results, the performance of the EDD-rule and the RR-rule was significantly worse compared to the SA and EDD per operation. The SA metaheuristic has the best performance, closely followed by the EDD per operation heuristic. The differences in performance between these two are negligible, but the computation time needed for the SA metaheuristic is a lot higher than for the EDD per operation heuristic. Thus, the EDD per operation heuristic seems to be the best method for improving the sequencing with the objective to minimize the tardiness and makespan of orders. According to the simulated results, the average service level would increase from 0.908 to 0.946 and the makespan would decrease from 74.4 hours to 67.42 hours. Besides, also the variance in the service level and makespan would decrease, resulting in better predictability and a lower impact of disruptions in the process.

10 Conclusion, implementation, and discussion

In this chapter, the conclusion is formulated by answering the main- and sub research questions. Furthermore, a plan for implementation for the recommended changes based on the conclusion is given. Lastly, the results of this research will be discussed and a direction for further research is given in the discussion.

10.1 Conclusion

The main research question in this thesis was: *How to improve the service level and the makespan of doors produced in the lacquered door factory of Berkvens?* This main research question was supported by several sub research questions (SRQ), which will be answered below. Based on these answers, the main research question will be answered.

The first SRQ is the following: *What are the parameters and performances of the individual operations and how do the operations interact with and influence each other?* Data has been gathered from several sources and analyses have been done on the performance of the operations. Furthermore, the demand, standstills, and repairs and discards have been investigated, and the flow between operations as well. The arrival rates and service rates per operation are investigated based on all the above factors, and it can be concluded that the operations in the LDF should be able to deal with the demand, as the utilization rates of all operations are between 0.7 and 0.9. This indicates that the system has a high load and disruptions in the process could have a long lasting effect, but overall the LDF should be able to deal with the demand.

The second SRQ is: *What distinction can be made between MTO and MTS orders regarding performance and required performance, and how should the differences between those be addressed in the sequencing of orders?* A clear distinction can be made between MTO and MTS orders, both in complexity, lead time, and order size. MTS orders make up approximately 30% of the total demand and have a standard production lead time of 20 days. The average size of the production orders for MTS doors seems bigger than the average size of the production orders for MTO doors, probably due to the higher degree of standardization. The configuration of the MTS doors is namely determined beforehand, whereas the configuration of the MTO doors is largely influenced by customer requirements. The weekly number of MTS doors produced depends on the demand for the MTO orders, as the remaining capacity of the LDF after the MTO orders are produced is used for the MTS orders. However, analyses of the arrival processes for the MTO- and MTS orders separately did not yield significant results, and thus no definite claims can be made on order sizes or number of orders per time period for MTO and MTS separately. When looking at the current sequencing rules, the MTS doors have lower priority, which results in a longer average makespan for MTS doors compared to MTO doors. This lower priority is logical, as the customer for MTS orders is Berkvens' warehouse, so late deliveries result in fewer negative consequences. Therefore, the MTO orders should be prioritized in the sequencing of orders.

The third SRQ is: *How is the due date for production orders determined and what adjustments can be made in this based on lead times?* The standard start date for production orders is five weeks before the product is due, which is usually three weeks after the sales order is made. In short: if an order is placed in week 0, the order is started in week 3, and the order is due in week 8. Adjustments can be made based on lead times and special circumstances. NLC orders have a standard delivery time of two weeks, with orders from two large customers being prioritized. Due dates can also be determined far in advance for large projects. Shifts in due dates occur frequently due to disruptions, and changes can happen even after the production order is released. Overall, no adjustments in the order due date can be made without changing the sales process, which is out of scope.

The fourth and last SRQ is: *How is the production sequence of jobs and batches determined and what improvements can be made in the sequencing with the objective to minimize the tardiness and makespan of orders?* Currently, the sequence of products and batches is determined based on the priority level of products and batches. Based on the type of order and the remaining slack time, a priority number is assigned to a product or batch. The operators at the machines can choose freely what product or batch to process next, as long as the products or batches with the highest priority are processed first. Multiple methods are developed and tested, and the most optimal way to determine the sequence is to prioritize the products or batches based on the Earliest Due Date (EDD) rule per operation. For this rule, a deadline for every operation the product or batch needs to visit is determined, based on the average waiting time per operation, the total remaining service time, and the slack of the product. Based on simulation results of this method, the average service level will increase from 0.908 to 0.946, the service level for MTO orders will increase from 0.881 to 0.916, and the average makespan will decrease from 74.4 to 67.4 hours.

Circling back to the main research question: the service level and makespan of doors produced in the lacquered door factory (LDF) of Berkvens can be improved by implementing the EDD sequencing rule per operation. By implementing this method, the average service level will increase from 90.8% to 94.6% and the average makespan will decrease from 74.4 hours to 67.4 hours. Furthermore, the service level of MTO orders will increase as well, from 88.1% to 91.6%, thereby reducing the amount of penalty's incurred as a result of late deliveries. Overall, it can be concluded that the LDF can improve its service level and makespan through appropriate sequencing of orders and production sequence planning.

10.2 Implementation

Before the new sequencing method can be implemented, a couple of steps should be taken. First of all, the new method should be explained and communicated with the stakeholders, which are the operators at the machines, as well as the managers of the production at the LDF, and the planners of the production orders. Specifically, the foreman that is in charge when determining the release of the batches, as described in Section 5.1, should be involved in this process. Only when all stakeholder understand the new method and agree that this method will improve the performance, the new sequencing method can be implemented successfully.

Currently, the operators have a free choice in what batch to process next as long as they adhere the priority rules. When the new method is implemented, the choice for the next batch is no longer arbitrarily, but is determined by the heuristic. This heuristic needs information what batches are currently available for production and some parameters of these batches to determine the optimal sequence. Currently, this data is not available, as the real-time location of batches is not registered in the ERP-system or any other system. Therefore, it is impossible to implement the new sequencing heuristic immediately. A new ERP-system will be implemented over the next year, with the target of going live on 1 January 2024. In this ERP-system, the location of products and batches should be registered, which can be used as input for the sequencing heuristic. Thus, only when the new ERP-system is implemented and data on the location and status of the products and batches is available, the new sequencing heuristic can be implemented.

When the new ERP-system is implemented and the data that is needed for determining the sequence is available, the new sequencing heuristic can be implemented. Based on the data on available products and batches, the optimal sequence can be determined. The determination of the optimal sequence can be done automatically, but this sequence needs to be communicated to the operators, which is a bigger challenge. This communication should be immediately clear to ensure a smooth and quick flow in the factory. The best way to communicate the new sequence is via screens, where the batch or product number is communicated together with the time the batch or product left the previous operation, such that the approximate location in the batch can be determined. This should ensure a clear communication and a good implementation of the new sequencing method.

10.3 Discussion

The key findings of the research conducted in this thesis are the following:

- The operations in the LDF should be able to deal with the demand;
- MTO orders should be prioritized in the sequencing of orders;
- No adjustments in the order due date can be made without changing the sales process;
- Prioritizing the products or batches based on the Earliest Due Date (EDD) rule per operation is the most optimal way to determine the sequence of products and batches. By implementing this method, the average service level will increase from 90.8% to 94.6% and the average makespan will decrease from 74.4 hours to 63.4 hours.

These key findings imply that the service level and makespan of the LDF can be improved. However, based on the results of the simulation model, it seems highly unlikely that the KPI of a 98% service level will be reached, as the upper bound of the 95% confidence interval stops at 95.6%. Another way to increase the service level would be to alter the sales process, such that adjustments in the determination of the due date for products can be realized. Furthermore, these key findings indicate that the EDD rule per operations performs similarly as the Simulated Annealing metaheuristic, which is a surprising result, as the literature indicated that SA performs better in general. This surprising performance can be explained by the choices of parameters for the SA, and an altered version of the SA metaheuristic could have a better performance.

The main limitation of this study was the lack of reliable data sources that were interconnected. Within the company, several different systems and programs are used to store information, and often those systems and programs are independent. Therefore, it was hard and time consuming to find reliable data and to connect data sources. In the end, the data used in this research is known to contain some errors and discrepancies from reality. However, it was the best data available and as reliable as possible. Besides the problems with the reliability of the data, no data was available for several processes, such as the drill street, which was new. Thus assumptions had to be made, which could influence the validity of the simulation model. However, the validity of the model has been tested, and overall it seems the model represents the reality fairly well, thus conclusions drawn based upon output of the simulation model seem reliable and valid.

In the literature review conducted on the topic of hybrid flowshop scheduling, many papers were considered that compared the performance of several heuristics and metaheuristics. Based on the conclusions of these papers, it was expected that the Earliest Due Date (EDD) dispatching rule would perform worse than the RR dispatching rule, which was not in line with the output of the simulation models used in this paper. Furthermore, the Simulated Annealing (SA) heuristic outperformed both the EDD-rule and the RR-rule, which was in line with the findings of the reviewed papers. However, when a deadline was determined per operation and the EDD-rule was implemented on this deadline per operation, the performance of the SA metaheuristic was similar to the performance of this EDD-rule per operation. As this deadline per operation was another dispatching rule, although adapted, the conclusion can be drawn that metaheuristics do not necessarily outperform heuristics. This might be an important conclusion for further research on the topic of the hybrid flowshop problem. Furthermore, the specific form of this hybrid flowshop problem might be interesting for further research as well, as this set up has not been researched before as far as the author is concerned. Thus the conclusion about the best method might be similar for similar set ups.

Further research on the topic of the hybrid flowshop scheduling problem with failures and maintenance, rework, limited buffer sizes, and overwork, could be executed. Then, the performance of the heuristics and metaheuristic used in this study could be evaluated with other parameters, and conclusions could be drawn based upon results from several studies.

Further research on improving the delivery reliability within Berkvens could be done to reach the KPI for the service level. One research direction could be analyzing the sales process and seeing if and what changes could be made in this process to allow better determination of due dates. Another research direction could be to analyze the cause of disruptions in the system and diminishing those. Also, the amount of rework could be reduced, such that the flow is less disrupted. Carrying out one or all of these proposals should result in an increased performance and thus in an increased service level.

References

[Aarts and Korst, 1989] Aarts, E. and Korst, J. (1989). *Simulated annealing and Boltzmann machines : a stochastic approach to combinatorial optimization and neural computing.* Wiley-Interscience series in discrete mathematics and optimization. Wiley, United States.

[Aurich et al., 2016] Aurich, P., Nahhas, A., Reggelin, T., and Tolujew, J. (2016). Simulation-based optimization for solving a hybrid flow shop scheduling problem. In *2016 Winter Simulation Conference (WSC)*, pages 2809–2819.

[Azadeh et al., 2019] Azadeh, A., Goodarzi, A. H., Kolaee, M. H., and Jebreili, S. (2019). An efficient simulation–neural network–genetic algorithm for flexible flow shops with sequence-dependent setup times, job deterioration and learning effects. *Neural Comput. Appl.*, 31(9):5327–5341.

[Baxendale et al., 2021] Baxendale, M., McGree, J. M., Bellette, A., and Corry, P. (2021). Machine-based production scheduling for rotomoulded plastics manufacturing. *International Journal of Production Research*, 59(5):1301–1318.

[Bayram and Şahin, 2013] Bayram, H. and Şahin, R. (2013). A new simulated annealing approach for travelling salesman problem. *Mathematical and Computational Applications*, 18(3):313–322.

[Behnamian et al., 2010] Behnamian, J., Zandieh, M., and Fatemi Ghomi, S. M. T. (2010). Due windows group scheduling using an effective hybrid optimization approach. *Int. J. Adv. Manuf. Technol.*, 46(5-8):721–735.

[Bernate Lara et al., 2013] Bernate Lara, A. F., Yalaoui, F., Dugardin, F., and Entzmann, F. (2013). Heuristic method for the hybrid flow shop problem in batch production. In *2013 International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 149–153.

[Boon et al., 2021] Boon, M., van der Boor, M., van Leeuwaarden, J., Mathijsen, B., van der Pol, J., and Resing, J. (2021). Stochastic simulation using python.

[Bozorgirad and Logendran, 2016] Bozorgirad, M. A. and Logendran, R. (2016). A comparison of local search algorithms with population-based algorithms in hybrid flow shop scheduling problems with realistic characteristics. *Int. J. Adv. Manuf. Technol.*, 83(5-8):1135–1151.

[Bramoullé and Saint-Paul, 2010] Bramoullé, Y. and Saint-Paul, G. (2010). Research cycles. *Journal of Economic Theory*, 145(5):1890–1920.

[CBS and Swagerman, 2023] CBS and Swagerman, A. (2023). Hoogste aantal nieuwbouwwoningen in afgelopen decennium. cbs.nl.

[Cokelaer, 2019] Cokelaer, T. (2019). Fitter documentation. Accessed 26-12-2022.

[Dowsland and Thompson, 2012] Dowsland, K. A. and Thompson, J. M. (2012). Simulated annealing. In *Handbook of Natural Computing*, pages 1623–1655. Springer Berlin Heidelberg.

[Fishman, 2013] Fishman, G. S. (2013). *Discrete-event simulation.* Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 2001 edition.

[Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and intractability.* W.H. Freeman, New York, NY.

[Glover and Laguna, 1998] Glover, F. and Laguna, M. (1998). *Tabu Search*, pages 2093–2229. Springer US, Boston, MA.

[Gupta, 1988] Gupta, J. N. D. (1988). Two-stage, hybrid flowshop scheduling problem. *The Journal of the Operational Research Society*, 39(4):359–364.

[Gurobi Optimization, LLC, 2023] Gurobi Optimization, LLC (2023). Gurobi Optimizer Reference Manual.

[Hopp and Spearman, 2011a] Hopp, W. J. and Spearman, M. L. (2011a). *Factory physics*. Waveland PressInc.

[Hopp and Spearman, 2011b] Hopp, W. J. and Spearman, M. L. (2011b). *Factory physics*. Waveland PressInc.

[Jeuken, 2022] Jeuken, S. (2022). Hybrid flow shop scheduling to increase delivery reliability.

[Jungwattanakit et al., 2008] Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., and Werner, F. (2008). Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. *Int. J. Adv. Manuf. Technol.*, 37(3-4):354–370.

[Khalouli et al., 2008] Khalouli, S., Ghedjati, F., and Hamzaoui, A. (2008). Hybrid approach using ant colony optimization and fuzzy logic to solve multi-criteria hybrid flow shop scheduling problem. In *Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology*, CSTST '08, page 44–50, New York, NY, USA. Association for Computing Machinery.

[Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

[Lee et al., 2011] Lee, C. K. M., Lin, D., Ho, W., and Wu, Z. (2011). Design of a genetic algorithm for bi-objective flow shop scheduling problems with re-entrant jobs. *Int. J. Adv. Manuf. Technol.*, 56(9-12):1105–1113.

[Li et al., 2012] Li, X., Yalaoui, F., Amodeo, L., and Chehade, H. (2012). Metaheuristics and exact methods to solve a multiobjective parallel machines scheduling problem. *J. Intell. Manuf.*, 23(4):1179–1194.

[Linn and Zhang, 1999] Linn, R. and Zhang, W. (1999). Hybrid flow shop scheduling: A survey. *Computers Industrial Engineering*, 37(1):57–61. Proceedings of the 24th international conference on computers and industrial engineering.

[Luke, 2013] Luke, S. (2013). *Essentials of Metaheuristics*. Lulu, second edition. Available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/.

[Mann, 2016] Mann, P. S. (2016). *Introductory Statistics*. John Wiley & Sons, Nashville, TN, 9 edition.

[Massey, 2012] Massey, F. J. (2012). The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78.

[Montgomery and Runger, 2018] Montgomery, D. C. and Runger, G. C. (2018). *Applied statistics and probability for engineers*.

[Mousavi et al., 2013] Mousavi, S. M., Mousakhani, M., and Zandieh, M. (2013). Bi-objective hybrid flow shop scheduling: a new local search. *Int. J. Adv. Manuf. Technol.*, 64(5-8):933–950.

[Nahmias and Olsen, 2015] Nahmias, S. and Olsen, T. (2015). *Production and operations analysis: Strategy, quality, analytics, application.*

[Portmann, 1996] Portmann, M.-C. (1996). Genetic algorithms and scheduling: a state of the art and some propositions. In *Proceedings of the Workshop on Production Planning and Control, pages i–xxiv, Mons (Belgium)*, volume 9.

[Raghu and Rajendran, 1993] Raghu, T. and Rajendran, C. (1993). An efficient dynamic dispatching rule for scheduling in a job shop. *International Journal of Production Economics*, 32(3):301–313.

[Rajendran and Holthaus, 1999] Rajendran, C. and Holthaus, O. (1999). A comparative study of dispatching rules in dynamic flowshops and jobshops. *European Journal of Operational Research*, 116(1):156–170.

[Ruiz and Vázquez-Rodríguez, 2010] Ruiz, R. and Vázquez-Rodríguez, J. A. (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1):1–18.

[Sáenz-Alanís et al., 2016] Sáenz-Alanís, C. A., Jobish, Salazar-Aguilar, M. A., and Boyer, V. (2016). A parallel machine batch scheduling problem in a brewing company. *Int. J. Adv. Manuf. Technol.*, 87(1-4):65–75.

[Sargent, 2011] Sargent, R. (2011). Verification and validation of simulation models. volume 37, pages 166 – 183.

[Schlesinger, 1979] Schlesinger, S. (1979). Terminology for model credibility. *Simulation*, 32(3):103–104.

[Sun et al., 2011] Sun, Y., Zhang, C., Gao, L., and Wang, X. (2011). Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. *Int. J. Adv. Manuf. Technol.*, 55(5-8):723–739.

[Tan et al., 2018] Tan, Y., Mönch, L., and Fowler, J. W. (2018). A hybrid scheduling approach for a two-stage flexible flow shop with batch processing machines. *J. Sched.*, 21(2):209–226.

[Ünal et al., 2020] Ünal, A. T., Ağralı, S., and Taşkın, Z. C. (2020). A strong integer programming formulation for hybrid flowshop scheduling. *J. Oper. Res. Soc.*, 71(12):2042–2052.

[Velleda Gonzales et al., 2015] Velleda Gonzales, G., Domingues dos Santos, E., Ramos Emmendorfer, L., André Isoldi, L., Oliveira Rocha, L. A., and Da Silva Diaz Estrada, E. (2015). A comparative study of simulated annealing with different cooling schedules for geometric optimization of a heat transfer problem according to constructal design. *Sci. Plena*, 11(8).

[Voß and Witt, 2007] Voß, S. and Witt, A. (2007). Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application. *International Journal of Production Economics*, 105(2):445–458. Scheduling in batch-processing industries and supply chains.

[Xiong et al., 2017] Xiong, H., Fan, H., Jiang, G., and Li, G. (2017). A simulation-based study of dispatching rules in a dynamic job shop scheduling problem with batch release and extended technical precedence constraints. *European Journal of Operational Research*, 257(1):13–24.

[Yalaoui et al., 2011] Yalaoui, N., Mahdi, H., Amodeo, L., and Yalaoui, F. (2011). A particle swarm optimization under fuzzy logic controller to solve a scheduling problem. In *2011 International Conference on Communications, Computing and Control Applications (CCCA)*, pages 1–6.

[Zhang and Chen, 2016] Zhang, X.-Y. and Chen, L. (2016). Heuristics for minimizing the total tardiness in a re-entrant hybrid flow shop with non-identical machines in parallel. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 987–991.

[Zhang and Chen, 2018] Zhang, X. Y. and Chen, L. (2018). A re-entrant hybrid flow shop scheduling problem with machine eligibility constraints. *International Journal of Production Research*, 56(16):5293–5305.

11 Appendix

11.1 Appendix 1: Potential problem causes

A cause-and-effect diagram has been made, where the effect is the low delivery reliability and long makespan as described in the previous section. Potential causes for this effect have been identified, by looking at four main areas: man, machine, material, and method. The diagram is displayed in Figure 34. The potential causes for the effect are explained in more detail below.

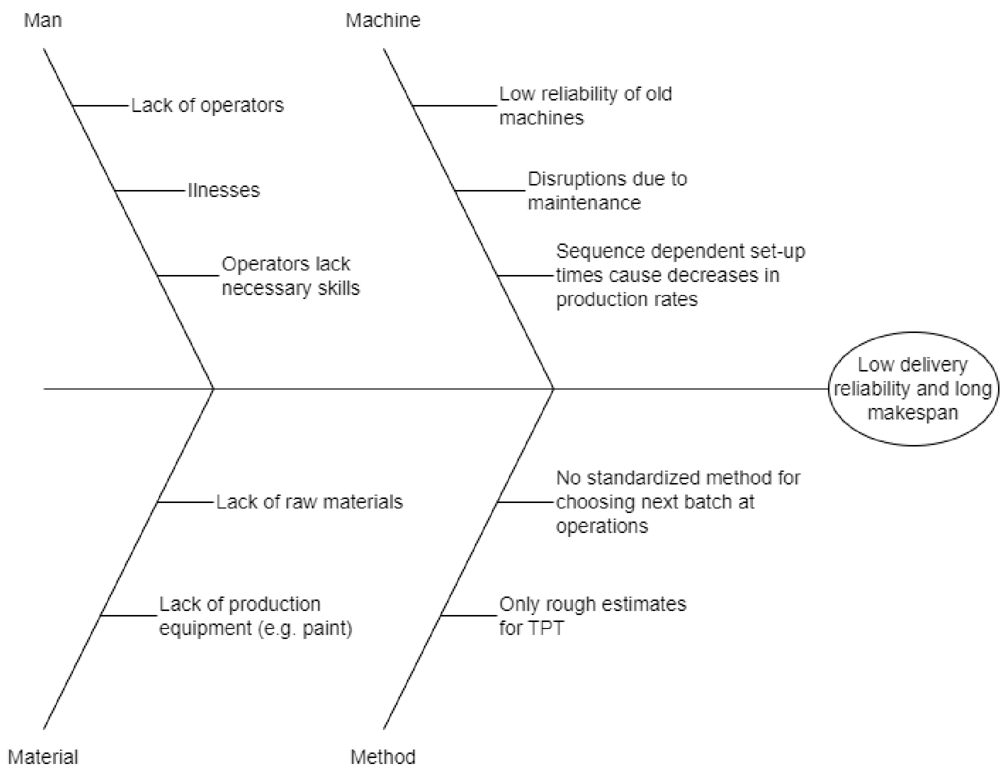


Figure 34: Cause-and-effect diagram for the problem

Man

- Lack of operators: A lack of operators may result in operations that cannot produce due to lack of necessary operators to run the operation, handle the materials, perform the quality check, etc. These effects lead to standstills in the operations, which causes longer makespans and late deliveries;
- Illnesses: illnesses can result in a lack of operators for given days or periods, of which the results are mentioned above;
- Operators lack necessary skills: Even when enough operators are available, some operators might lack the skills necessary for performing their duties to enable the process to run smoothly. This can result in the effects stated above.

Machine

- Low reliability of old machines: Some machines have low reliability and thus many unpredictable disruptions, which cause an increase in the makespan, and might cause orders to be tardy;

- Disruptions due to maintenance: maintenance should be done on certain machines. Most of the time, the maintenance is planned during the nights, weekends, and/or holidays. However, sometimes maintenance activities need to be done during production hours, which causes an increase in the makespan due to increased waiting times and possible delayed orders resulting in a low delivery reliability;
- Sequence-dependent setup times: Some operations have sequence-dependent setup times, which can cause decreases in the production rate if a sub-optimal sequence is chosen. A decrease in the production rate causes longer waiting time, with the effects as mentioned before.

Material

- Lack of raw materials: A lack of raw materials can result in a production stop because the required materials to produce doors are unavailable. This results in a longer makespan and hence perhaps reduced delivery reliability;
- Lack of production equipment: A lack or production equipment might for instance be paint, drills, cardboard, etc. This can cause disruptions in the processes, with the aforementioned effects as a result.

Method

- No standardized method for choosing next batch at operations: operators have the freedom to choose themselves what batch they want to produce next on their operations. They should adhere to priority rules, but furthermore they have freedom in choosing batches, which can cause some batches to have a long waiting time before an operation, while others can go quickly through. This causes a lot of variability in the makespan, which is therefore harder to predict and its estimates should be increased to overcome this variability, resulting in a longer makespan overall;
- Only rough estimates for TPT: there are only rough theoretical estimates for the throughput time, which results in more products in WIP, which causes longer waiting times and a longer makespan.

11.2 Appendix 2: Results distribution fit production times per product

In Figure 35 until 40 the results of a distribution fit to data about the production times per product are displayed. In these figures, the probability distributions with the best fit for the data are given.

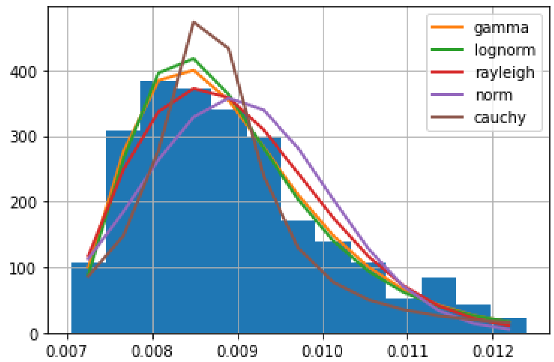


Figure 35: Hourly production time per product mass press

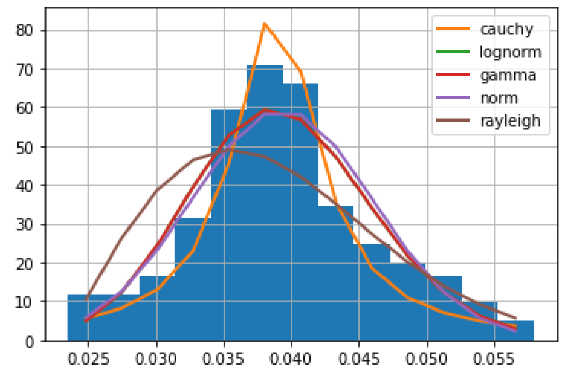


Figure 36: Hourly production time per product small series press

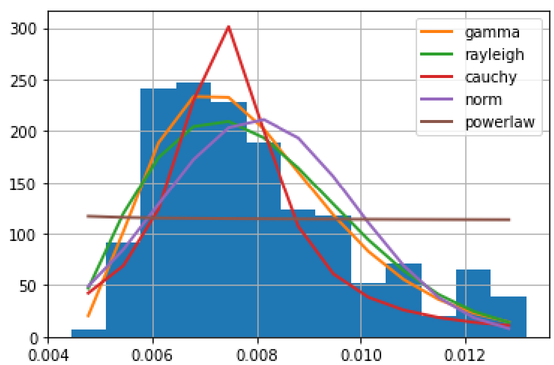


Figure 37: Hourly production time per product edge processing

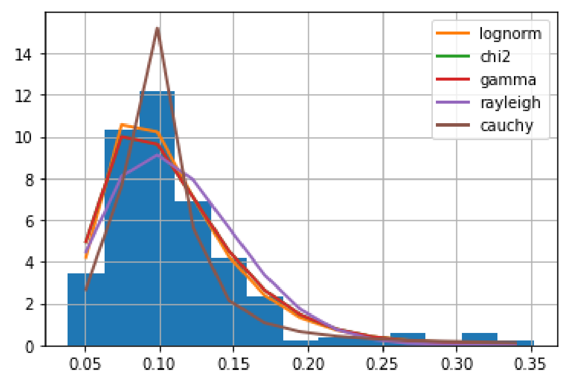


Figure 38: Hourly production time per product CNC

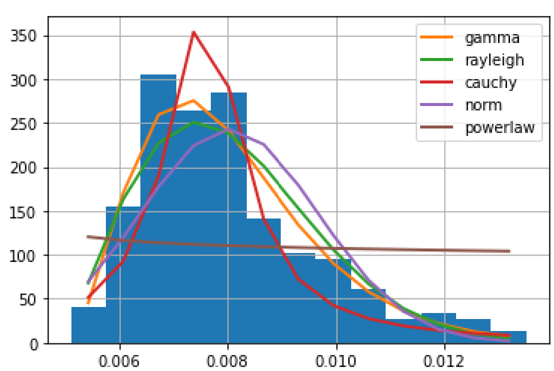


Figure 39: Hourly production time per product Painting

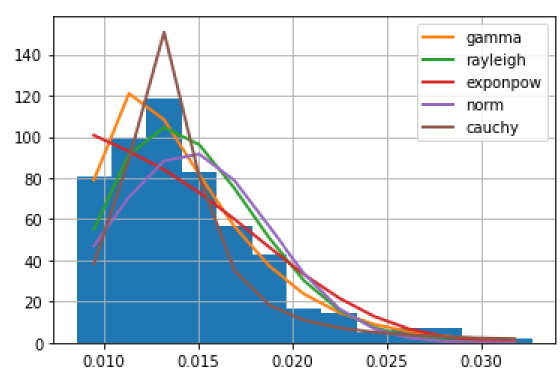


Figure 40: Hourly production time per product Packing

In Table 32 the results of a Kolmogorv-Smirnov test for goodness of fit for the production times per product for the main operation are given. Here, the null hypothesis (h_0) is as follows: the production time per product follows the Normal distribution. Based on the results in this table, it can be concluded that the null hypothesis is rejected for all operations except the two presses, which means that only the production time per product may well follow the Normal distribution. The other operations do not seem to follow the normal distribution. However, based on the results obtained with the fitter, displayed in the figures above, the Gamma distribution seems to have a fairly good fit with all operations. Therefore, the goodness of fit test is repeated with the Gamma distribution as well. The null hypothesis (h_0) is then: the production time per product follows the Gamma distribution. The results of these tests are given in Table 33, and the null hypothesis are accepted in all cases.

Operation	**Mean (μ)**	**StDev (σ)**	**P-value (norm dist)**	h_0
Mass press	0.0089	0.0011	0.0680	Accepted
Small series press	0.0394	0.0067	0.3808	Accepted
Edge processing	0.0080	0.0019	0.0119	Rejected
CNC	0.1101	0.0490	0.0006	Rejected
Painting	0.0080	0.0016	0.0000	Rejected
Packing	0.0145	0.0045	0.0009	Rejected

Table 32: Results goodness of fit test Normal distribution production times per product

Operation	**Shape**	**Scale**	**Loc**	**P-value (gamma dist)**	h_0
Mass press	3.62	0.0006	0.0068	0.9823	Accepted
Small series press	91.62	0.0007	-0.0251	0.6786	Accepted
Edge processing	4.30	0.0009	0.0041	0.5799	Accepted
CNC	2.96	0.0264	0.0319	0.1414	Accepted
Painting	3.85	0.0008	0.0048	0.6442	Accepted
Packing	2.05	0.0030	0.0084	0.9732	Accepted

Table 33: Results goodness of fit test Gamma distribution production times per product

11.3 Appendix 3: System of equations for arrivals per operation

$$E(x_1) = E(x_2) \tag{25}$$

$$E(x_2) = E(x_3) + E(x_4) \tag{26}$$

$$E(x_3) = 0.189 \cdot 1527.42 + 0.001 \cdot 0.189 \cdot E(x_5) + 0.004 \cdot 0.189 \cdot (E(x_6) + E(x_7)) + 0.009 \cdot 0.189 \cdot E(x_8) + 0.017 \cdot 0.189 \cdot E(x_{10}) + 0.003 \cdot 0.189 \cdot E(x_{11}) \tag{27}$$

$$E(x_4) = 0.811 \cdot 1527.42 + 0.001 \cdot 0.811 \cdot E(x_5) + 0.004 \cdot 0.811 \cdot (E(x_6) + E(x_7)) + 0.009 \cdot 0.811 \cdot E(x_8) + 0.017 \cdot 0.811 \cdot E(x_{10}) + 0.003 \cdot 0.811 \cdot E(x_{11}) \tag{28}$$

$$E(x_5) = E(x_3) + E(x_4) + 0.004 \cdot E(x_5) \tag{29}$$

$$E(x_6) = 0.988 \cdot E(x_5) + 0.207 \cdot E(x_6) \tag{30}$$

$$E(x_7) = 0.003 \cdot E(x_5) + 0.1 \cdot E(x_6) + 0.03 \cdot E(x_7) \tag{31}$$

$$E(x_8) = 0.004 \cdot E(x_5) + 0.689 \cdot E(x_6) + 0.966 \cdot E(x_7) + 0.025 \cdot E(x_8) \tag{32}$$

$$E(x_9) = 0.031 \cdot E(x_8) \tag{33}$$

$$E(x_{10}) = 0.158 \cdot E(x_8) + 0.123 \cdot E(x_9) + 0.028 \cdot E(x_{10}) \tag{34}$$

$$E(x_{11}) = 0.738 \cdot E(x_8) + 0.877 \cdot E(x_9) + 0.913 \cdot E(x_{10}) + 0.004 \cdot E(x_{11}) \tag{35}$$

11.4 Appendix 4: Description classes used in simulation

Product class

The product class keeps track of all the products. Input parameters are the arrival time, priority level, due time, route, and service times. Furthermore, each product has some parameters that are relatively stable, namely: a product number and the batch number of the batch that it is assigned to. Then there are some parameters that are updated more often, namely: the location of the product, the status of sawing and labelling activities, the *SawLabComplete* parameter, which is *False* when a product is initiated, and the system depart time. The Product object also has some methods, namely: the *checkCompletenessSawLab* method, which checks if all sawing and labelling activities are completed and changes the *SawLabComplete* parameter to True when this is the case, the *batch* method, which assigns a batch number, the *moveTo* parameter, which updates the location of a product, and the *leaveSystem* method, which changes the location of the product to finished.

Batch class

The batch class keeps track of all the batches. Input parameters are the priority level, due time, and route of a batch. Furthermore, each batch has some parameters that are relatively stable, namely: a list of products that are assigned to the batch, an arrival time, the service times at each operation, and the number of stacks. Then there are some parameters that are updated more often, namely: the times the Drill/CNC/SPEC operation are completed and the current location of the batch. The Batch object also has some methods, namely: the *appendProduct* method, which adds a product to the batch, the *DetermineServiceTimes* method, which generates random sample times from the service distributions and assigns them to the batch, the *checkCompletePress* method, which checks if all products in the batch are ready for the press, the *moveTo* method, which updates the location of the batch, and the *leaveSystem* method, which changes the location of the batch to finished.

Event class

The event class stores details about all the events. Its parameters are the following: the event type, which can be either *Arrival* or *Departure*, the time of the event, the batch of the event, the product of the event, the priority level of the event, the location of the event, and the service time of the event, which is only bigger than 0 for arrival events. The Event class has one method, which is the built in method __*lt*__, which orders the events by the time parameter.

FES class

The FES class contains all sorted event objects, and this list of events is its only parameter. It has three methods, namely: the *add* method, which adds an event to the list, the *next* method, which returns the next event based on the times, and the *isEmpty* method, which returns "True" when there no more events.

NetworkSimulation class

In this class, the main simulation is executed. It has three parameters, namely: the arrival distribution, the service distribution and the number of servers. It has six methods, namely: the *simulate* method, in which the simulation is executed, this method will be explained further below, the *DetermineRouteSawLab* method, which determines which sawing/labelling activities the products flow through, the *DetermineRouteMains* method, which determines which main operations the products flow through, the *Batching* method, which forms batches from a list of products, the *ArrivalsSawingLabelling* method, which makes arrival events for the sawing and labelling activities, and the *DetermineBreakdowns* method, which generates samples of days with breakdowns and the duration of breakdowns and turns this into a list.

NetworkSimResults class

The NetworkSimResults class keeps track of results of the simulation and returns performance measures. It has several parameters, namely: a list with the results of each individual queue that it gathers from the SimResults class that will be described in more detail below, the time of the simulation, the number of queues in the simulation, the results of the individual batches, the warm-up period, a list of all the makespans, the number of products, and the number on time parameter, which tracks the number of products delivered on time. The NetworkSimResults class has several methods, namely: the *registerOnTime* method, which registers whether a product is finished before its due date, the *serviceLevel* method, which calculates the internal service level, the *registerMakespan* method, which registers the makespan of a product, the *averageMakespan* method, which calculates the average makespan, the *registerQueueLength* method, which registers the queue length of a specific queue for a given time, the *getMeanQueueLengths* method, which calculates the mean queue lengths of each queue, the *registerWaitingTime* method, which registers the waiting time per queue, the *getMeanWaitingTime* method, which calculates the mean waiting time per operation, the *registerBusyTime* method, which registers the time an operation runs uninterrupted, and the *getMeanBusyTime* method, which calculates the mean busy time.

SimResults class

The SimResults class is similar to the NetworkSimResults class, but this class keeps track of the results of individual queues as opposed to the results of the whole network. Thus all methods of the NetworkSimResults class that are calculated for individual queues, such as the mean queue length and the mean waiting time, are registered in this class and retrieved by the NetworkSimResults class.

11.5 Appendix 5: Algorithms used in simulation model

Algorithm 3 New arrivals

```
if New day has arrived then
    Generate number of arrivals from arrival distribution (Gamma dist.)
    for All arrivals do
        Determine priority with probabilities
        Determine the route through the operations with Batch flow
        Determine the route through the operations with Product flow
        Determine the due date of the product based on the priority and the route
        Determine the service time for all operations for individual flow
        Initialize product object in the Product Class
        Add the product to the PoolForBatching list, i.e. the products that need to be batched
        Determine the sawing and labelling activities the product
        Make arrival events for all sawing and labelling activities for the product
    end for
end if
```

Algorithm 4 Batching

```
if New product instances are made then
    Add products in the ProductsToBeScheduled list to the PoolForBatching list
    for All products in PoolForBatching do
        if Batch with corresponding due date, priority, route, and start time already exists
              AND batchSize < 96 then
            Add product the corresponding batch
        else
            Make new batch object in Batch class with due date, priority, route, and start time
            Add batch to the new batch object
            Add batch to the newBatches list
        end if
    end for
    for Batch in newBatches do
        Determine the service times for all operations with batch flow
        Determine how many stacks a batch consists of
    end for
end if
```

Algorithm 5 Product flow: Arrival Event

```
if event.type == Arrival AND event.location ≤ 3 then
    Add the product to the queue for this location
    Sort the queue for this location, from highest priority to lowest
    if Length queue ≤ number of servers then
        Register the waiting time
        Make a departure event for the product for the current time plus the service time
        Add the event to the FES
    end if
end if
```

Algorithm 6 Product flow: Departure Event

```
if event.type == Departure AND event.location ≤ 3 then
    Remove the product from the queue
    Adjust the status of the corresponding sawing/labelling activity
    if event.location == 0 then
        Make an arrival event for the product for the current time at the labelling queue
        Add the event to the FES
    else
        Check whether all sawing and labelling activities are done with
        product.checkCompletenessSawLab()
        if product.SawLabComplete == True then
            Check if all products are ready for the press with batch.checkCompletePress()
            if batch.checkCompletePress() == True then
                Register the waiting time
                Determine the next queue based on the route
                Determine the next service time based on the route
                Make an arrival event for the batch for the current time at the next queue
                with the next service time
                Add the event to the FES
            end if
        end if
    end if
    if Length queue ≥ number of servers at current location then
        Determine next product in queue
        Register the waiting time
        Determine the next service time of this product
        Make a departure event for the product for the current time plus the service time
        Add the event to the FES
    end if
end if
```

Algorithm 7 Batch flow: Arrival Event

```
if event.type == Arrival AND event.location ≥ 4 then
    Add the product to the queue for this location
    Sort the queue for this location, from highest priority to lowest
    Add the number of stacks the batch consists of to the buffer
    if BufferSize < Stacks in buffer then
        Block preceding machine
    end if
    if Length queue ≤ number of servers AND statusOperation[event.location] == 0 then
        Register the waiting time
        Make a departure event for the product for the current time plus the service time
        Add the event to the FES
    else if Length queue ≤ number of servers AND statusOperation[event.location] == 1 then
        Add batch to the list of products that is waiting for the block to be over
    end if
end if
```

Algorithm 8 External operation

```
if NextOperation == External then
    Appended batch to queueExternal list
end if
if day of the week == 2 then
    All batches in the list queueExternal are sent out
    Make departure event for 7 days later for all batches that are sent out
    Add events to the FES
    Empty queueExternal
end if
```

Algorithm 9 Determine breakdowns

```
Make an array with zeros for every day in the length of the simulation for storing the start times of the breakdown
Make an array with zeros for every day in the length of the simulation for storing the end times of the breakdown
for Day in range(nrDays) do
    Sample a number between 0 and 1 from the uniform distribution
    if number < ProbMachineBreakdown[prob day with standstill][machine] then
        Sample the length of the breakdown with exponential distribution and machine parameters
        Determine the start- and end time of the breakdown with the uniform distribution
        Replace the value in the list for the start times for the given day with the start time
        Replace the value in the list for the end times for the given day with the end time
    end if
end for
```

11.6 Appendix 6: Snippet output file simulation

ProdNr	Batch	Prio	ArrivalT	DueT	DepartT	Makespan	OnTime	ServiceT	WaitingT
17110	1687	1	288	408	327.52	39.52	1	1.22	38.30
17111	1698	3	288	504	319.68	31.68	1	0.96	30.72
17112	1685	3	288	504	319.68	31.68	1	0.95	30.73
17113	1693	2	288	384	319.68	31.68	1	0.93	30.76
17114	1689	4	288	768	319.93	31.93	1	0.92	31.01
17115	1691	3	288	504	319.68	31.68	1	0.93	30.76
17116	1662	3	288	504	319.93	31.93	1	2.21	29.71
17117	1669	4	288	768	319.46	31.46	1	2.41	29.05
17118	1656	3	288	504	322.85	34.85	1	14.09	20.76
17119	1685	3	288	504	319.68	31.68	1	0.93	30.75
17120	1683	4	288	768	319.68	31.68	1	0.93	30.76
17121	1681	2	288	480	544.76	256.76	0	13.05	243.70
17122	1687	2	288	384	319.93	31.93	1	0.96	30.96
17123	1700	2	288	384	319.68	31.68	1	0.94	30.74
17124	1681	2	288	480	544.76	256.76	0	13.07	243.69
17125	1687	2	288	384	319.93	31.93	1	1.14	30.78
17126	1693	2	288	384	319.68	31.68	1	0.94	30.75
17127	1669	4	288	768	319.46	31.46	1	2.42	29.04
17128	1687	2	288	384	319.93	31.93	1	0.93	31.00
17129	1690	4	288	768	319.68	31.68	1	0.93	30.76

Table 34: Snippet output simulation

11.7 Appendix 7: Additional simulation output current sequencing method

Operation	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Fillings Saw	8.30	7.76	5.53	11.08
Shortening Saw	8.73	8.48	5.70	11.76
Plate Saw	11.02	10.37	7.31	14.74
Labelling	18.09	25.52	8.96	27.22
SSP	12.57	263.41	0.00	106.82
MP	13.43	263.82	0.00	107.83
EP	76.62	608.81	0.00	294.48
Drill	99.10	684.24	0.00	343.95
CNC	161.80	990.46	0.00	516.23
Painter	167.21	984.91	0.00	519.65
Extern	-	-	-	-
SPEC	331.60	1369.51	0.00	821.66
Packer	31.42	417.38	0.00	180.77

Table 35: Simulation results waiting time per operation current method

Operation	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Fillings Saw	33.41	77.91	5.53	61.29
Shortening Saw	66.08	149.96	12.42	119.74
Plate Saw	59.28	117.00	17.41	101.15
Labelling	1289.32	1720.25	673.75	1904.89
SSP	0.47	1.40	0.00	0.96
MP	1.74	4.03	0.30	3.18
EP	9.63	19.90	2.51	16.75
Drill	5.53	13.55	0.68	10.38
CNC	3.36	17.92	0.00	9.77
Painter	20.67	57.21	0.20	41.14
Extern	-	-	-	-
SPEC	4.85	73.79	0.00	31.25
Packer	17.21	29.11	6.79	27.62

Table 36: Simulation results queue length per operation current method

Operation	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Fillings Saw	8.87	176.73	0.00	72.11
Shortening Saw	6.65	151.28	0.00	60.79
Plate Saw	11.22	180.24	0.00	75.72
Labelling	20.51	218.92	0.00	98.85
SSP	3.01	108.34	0.00	41.78
MP	1.08	62.06	0.00	23.28
EP	2.20	81.14	0.00	31.23
Drill	2.27	83.45	0.00	32.13
CNC	0.84	62.04	0.00	23.05
Painter	2.11	77.35	0.00	29.79
Extern	-	-	-	-
SPEC	36.03	352.74	0.00	162.26
Packer	1.50	70.04	0.00	26.56

Table 37: Simulation results working period per operation current method

11.8 Appendix 8: Additional simulation output EDD rule

Operation	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Fillings Saw	7.95	5.89	5.85	10.06
Shortening Saw	8.35	6.26	6.11	10.59
Plate Saw	10.40	8.17	7.48	13.32
Labelling	14.57	20.26	7.32	21.82
SSP	14.83	263.01	0.00	108.95
MP	16.97	279.12	0.00	116.85
EP	61.62	538.48	0.00	254.31
Drill	99.41	696.49	0.00	348.64
CNC	0.03	0.41	0.00	0.18
Painter	10.40	184.18	0.00	76.30
Extern	-	-	-	-
SPEC	154.70	853.32	0.00	460.05
Packer	2.77	67.52	0.00	26.93

Table 38: Simulation results waiting time per operation EDD rule

Operation	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Fillings Saw	50.83	86.75	19.78	81.87
Shortening Saw	100.92	168.01	40.80	161.04
Plate Saw	90.28	127.19	44.76	135.79
Labelling	1490.55	1362.64	1002.94	1978.15
SSP	0.73	1.85	0.07	1.39
MP	2.11	4.20	0.61	3.61
EP	10.50	15.66	4.89	16.10
Drill	4.99	8.31	2.02	7.97
CNC	0.03	0.58	0.00	0.23
Painter	10.59	20.65	3.20	17.98
Extern	-	-	-	-
SPEC	1.25	3.29	0.07	2.43
Packer	8.12	15.15	2.70	13.54

Table 39: Simulation results queue length per operation EDD rule

Operation	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Fillings Saw	3.49	24.61	0.00	12.29
Shortening Saw	2.53	24.69	0.00	11.36
Plate Saw	4.81	45.54	0.00	21.11
Labelling	7.69	1013.37	0.00	370.31
SSP	0.86	6.48	0.00	3.17
MP	0.33	2.83	0.00	1.34
EP	0.91	10.04	0.00	4.50
Drill	0.83	4.28	0.00	2.36
CNC	0.04	0.33	0.00	0.15
Painter	0.69	8.38	0.00	3.69
Extern	-	-	-	-
SPEC	10.59	1262.44	0.00	462.33
Packer	0.50	3.61	0.00	1.79

Table 40: Simulation results working period per operation EDD rule

11.9 Appendix 9: Additional simulation output RR rule

Operation	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Fillings Saw	8.53	10.47	4.79	12.28
Shortening Saw	8.91	10.43	5.18	12.64
Plate Saw	11.30	14.18	6.22	16.37
Labelling	18.40	28.52	8.19	28.60
SSP	41.04	443.85	0.00	199.86
MP	16.55	283.40	0.00	117.96
EP	139.22	867.45	0.00	449.63
Drill	170.08	984.74	0.00	522.46
CNC	628.08	1900.73	0.00	1308.24
Painter	429.61	1605.66	0.00	1004.17
Extern	-	-	-	-
SPEC	404.96	1488.33	0.00	937.54
Packer	5.77	87.57	0.00	37.11

Table 41: Simulation results waiting time per operation RR rule

Operation	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Fillings Saw	56.27	94.84	22.33	90.21
Shortening Saw	112.77	182.13	47.60	177.95
Plate Saw	98.78	136.42	49.97	147.60
Labelling	2444.49	2323.79	1612.95	3276.03
SSP	0.84	2.13	0.07	1.60
MP	2.75	5.15	0.90	4.59
EP	18.30	34.80	5.85	30.76
Drill	11.00	20.33	3.72	18.27
CNC	30.66	76.87	3.15	58.17
Painter	75.42	143.64	24.03	126.82
Extern	-	-	-	-
SPEC	4.90	16.41	0.00	10.78
Packer	19.62	25.93	10.34	28.90

Table 42: Simulation results queue length per operation RR rule

Operation	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Fillings Saw	3.52	5.10	1.70	5.35
Shortening Saw	2.56	4.78	0.85	4.26
Plate Saw	5.31	8.89	2.13	8.49
Labelling	15.30	53.85	0.00	34.57
SSP	0.96	2.94	0.00	2.02
MP	0.37	1.96	0.00	1.07
EP	1.06	4.55	0.00	2.68
Drill	1.16	4.03	0.00	2.61
CNC	0.27	7.30	0.00	2.88
Painter	1.11	9.68	0.00	4.57
Extern	-	-	-	-
SPEC	14.77	46.55	0.00	31.43
Packer	0.62	2.37	0.00	1.47

Table 43: Simulation results working period per operation RR rule

11.10 Appendix 10: Additional simulation output EDD rule per operation

Operation	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Fillings Saw	8.16	6.25	5.92	10.39
Shortening Saw	8.56	6.74	6.15	10.97
Plate Saw	10.79	9.01	7.57	14.02
Labelling	16.10	23.42	7.72	24.48
SSP	14.30	237.81	0.00	99.40
MP	20.60	302.97	0.00	129.01
EP	85.82	658.17	0.00	321.34
Drill	116.63	756.29	0.00	387.26
CNC	0.17	1.42	0.00	0.67
Painter	8.40	158.06	0.00	64.96
Extern	-	-	-	-
SPEC	206.63	1120.58	0.00	607.62
Packer	4.96	80.31	0.00	33.70

Table 44: Simulation results waiting time per operation EDD per operation rule

Operation	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Fillings Saw	52.90	91.19	20.27	85.54
Shortening Saw	105.46	175.75	42.57	168.35
Plate Saw	93.71	133.20	46.04	141.37
Labelling	1994.43	1842.29	1335.19	2653.67
SSP	0.66	1.59	0.09	1.23
MP	2.93	5.33	1.03	4.84
EP	11.46	16.47	5.57	17.36
Drill	6.50	10.30	2.82	10.19
CNC	0.12	1.31	0.00	0.59
Painter	14.13	28.96	3.77	24.49
Extern	-	-	-	-
SPEC	3.91	12.49	0.00	8.38
Packer	11.70	18.04	5.24	18.15

Table 45: Simulation results queue length per operation EDD per operation rule

Operation	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Fillings Saw	3.51	8.92	0.32	6.70
Shortening Saw	2.61	7.72	0.00	5.37
Plate Saw	5.43	11.83	1.20	9.67
Labelling	11.31	44.07	0.00	27.08
SSP	0.95	7.03	0.00	3.47
MP	0.40	4.21	0.00	1.90
EP	1.00	5.62	0.00	3.01
Drill	0.93	5.06	0.00	2.74
CNC	0.08	3.47	0.00	1.32
Painter	0.78	5.44	0.00	2.73
Extern	-	-	-	-
SPEC	11.07	36.47	0.00	24.12
Packer	0.54	3.38	0.00	1.74

Table 46: Simulation results working period per operation EDD per operation rule

11.11 Appendix 11: Additional simulation output simulated annealing

Operation	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Fillings Saw	7.97	5.65	5.95	9.99
Shortening Saw	8.33	6.35	6.06	10.60
Plate Saw	10.56	8.36	7.57	13.55
Labelling	11.53	16.16	5.74	17.31
SSP	14.73	291.72	0.00	119.12
MP	7.66	196.58	0.00	78.00
EP	69.09	592.84	0.00	281.24
Drill	112.78	765.93	0.00	386.86
CNC	17.03	184.85	0.00	83.18
Painter	63.01	462.22	0.00	228.41
Extern	-	-	-	-
SPEC	245.31	1029.66	0.00	613.76
Packer	9.19	101.08	0.00	45.36

Table 47: Simulation results waiting time per operation RR rule

Operation	Average	St. Dev.	95% Confidence Interval	
			Lower bound	Upper bound
Fillings Saw	38.96	80.70	10.08	67.84
Shortening Saw	77.00	154.79	21.61	132.39
Plate Saw	69.55	120.82	26.32	112.79
Labelling	859.18	1028.20	491.25	1227.11
SSP	0.48	1.21	0.04	0.91
MP	2.00	5.11	0.17	3.83
EP	9.58	20.52	2.24	16.92
Drill	5.81	13.93	0.83	10.80
CNC	1.39	11.27	0.00	5.42
Painter	18.70	47.30	1.78	35.63
Extern	-	-	-	-
SPEC	6.45	23.37	-1.91	14.81
Packer	11.69	23.47	3.29	20.09

Table 48: Simulation results queue length per operation RR rule

Operation	**Average**	**St. Dev.**	**95% Confidence Interval**	
			Lower bound	**Upper bound**
Fillings Saw	6.27	6775.36	0.00	2430.76
Shortening Saw	4.63	4926.05	0.00	1767.36
Plate Saw	7.92	7357.74	0.00	2640.81
Labelling	9.24	6800.87	0.00	2442.85
SSP	1.93	2786.09	0.00	998.90
MP	0.67	852.42	0.00	305.70
EP	1.52	1437.08	0.00	515.76
Drill	1.47	1441.74	0.00	517.38
CNC	0.42	832.87	0.00	298.45
Painter	1.41	1268.70	0.00	455.40
Extern	-	-	-	-
SPEC	61.41	67361.84	0.00	24166.09
Packer	0.94	1009.42	0.00	362.15

Table 49: Simulation results working period per operation RR rule