Design of a Method to Improve the Customer Service Level at an Electronics Production Company

Incorporating resource capacities using data available in tactical production planning

Master Thesis Industrial Engineering & Management

M.S. Zwanenburg

Department of Industrial Engineering & Business Information Systems
University of Twente

August 29, 2023

This report is intended for Hortec Assemblies BV and the supervisors of the University of Twente.

Design of a Method to Improve the Customer Service Level at an Electronics Production Company
Incorporating resource capacities using data available in tactical production planning

University of Twente
Faculty of Behavioral Management and Social Sciences
Industrial Engineering and Business Information Systems

Final report Master Thesis
Industrial Engineering & Management
Specialisation: Production & Logistics Management

Author: M.S. Zwanenburg

August 29, 2023

Supervisors of University of Twente:
Dr. E. Topan
Faculty of Behavioral Management and Social Sciences
Industrial Engineering and Business Information Systems

Dr. Ir. J.M.J. Schutten
Faculty of Behavioral Management and Social Sciences
Industrial Engineering and Business Information Systems

Supervisor of Hortec Assemblies BV:
L.A. Zwanenburg
Director of Hortec Assemblies BV

Edition: 1
Number of pages: 60
Number of appendices: 1

Management summary

Introduction and problem description

This research is conducted at Hortec Assemblies BV (HA), a company that produces and assembles (control) electronics for third parties operating in aerospace, railway, automotive and industry. Hortec employs a high-mix low-volume Make-To-Order production strategy in a job shop production environment with 2 production departments: Surface-Mount Technology (SMT) and Through-Hole Technology (THT). The SMT department performs 3 distinct operations and the THT department performs 14 distinct operations. Production planning at HA for both departments predominantly relies on the experience, assumptions and gut feelings of the operations manager. Management at HA finds its current Customer Service Level of 86.6% too low and wants to increase it to 90%. After discussion with management, we create a problem cluster to identify possible (core) problems. The low Customer Service Level results from end products missing their external due dates for customer shipments. This issue arises from infeasible production plans, driven by unrealistic external and internal due dates and production plans that rely on assumptions about resource capacity instead of incorporating actual resource capacity data. Therefore, the core problem of this research is the lack of incorporating resource capacities in tactical production planning using the data available. The main research question is:

How can a method that takes into account resource capacity using data available in tactical production planning for a high-mix low-volume make-to-order EMS company be designed such that the Customer Service Level improves from 86.6% to 90%?

Approach and solution design

HA lacks differentiation between strategic, tactical and operational planning. To limit the scope, we use a positioning framework that categorises different capacity planning functions such as order acceptance, resource loading and scheduling. Order accepting decides on accepting or rejecting incoming orders. Resource loading measures the impact of a set of orders on the production system, and determines reliable due dates and resource capacity levels needed to produce the orders and their constituting jobs. Scheduling is about the assignment of jobs to machines. For HA, resource loading is most appropriate since we incorporate resource capacity levels in tactical production planning such that the last confirmed due date is achieved and the Customer Service Level increases.

Resource loading focus on a time-driven approach, resource-driven approach or a combination of both. The time-driven approach extends short-term capacity at a certain cost to ensure timely order completion. The resource-driven approach cannot extend capacity and allows orders to finish after their due date at a certain cost. For HA, we design a resource loading MIP model that integrates both approaches. The MIP model takes into account (1) non-preemption constraints within the operation of an order, (2) capacity constraints and (3) precedence constraints. This implies scheduling operations within one or multiple consecutive periods without exceeding capacity limits while maintaining the predecessor-successor relationship. The MIP model is computationally tractable for smaller data instances.

For larger data instances, we use heuristic methods. These methods first create an initial solution using a constructive heuristic, which is then improved using an improvement heuristic. For the constructive heuristic, we find priority rules and finite loading methods to construct an initial solution. For HA, we designed a constructive heuristic based on Partial Backwards Finite Loading and priority rules. For the priority rules, we test Earliest Due Date, Latest Due Date, Shortest Processing Time and Latest Processing Time. For the improvement heuristic of HA, we use the metaheuristic Simulated Annealing.

Results and conclusion

We assess the MIP model and heuristics for a planning horizon of 1, 2, 4 or 6 months considering light, typical and heavy workloads. The designed MIP model is computationally tractable for a planning horizon of 1 or 2 months in combination with a light, typical and heavy workload. The Customer Service Level can improve beyond 90% at the expense of tardiness and overtime costs. For the scenarios tested, the Customer Service Level can improve between 95% and 100% with costs ranging between €0 and €50,500. The bottleneck operations for HA are SMD, SSOL, AOI and PR. For a planning horizon larger than 2 months the MIP model becomes computationally expensive and the heuristics are recommended to use. The constructive heuristic performs best using the Shortest Processing Time priority rule. The heuristics achieve gaps between 24-35.2% from the MIP solution. Reasons for the high gaps are (1) a constructive heuristic that does not decide on overtime and tardiness and (2) an improvement heuristic that finds a current best solution quickly but is not able to improve the solution more.

Recommendations and implementation challenges

Recommendations for HA are (1) implementing the MIP model at HA, (2) further developing the constructive and improvement heuristic, (3) implementing the heuristics in a more suitable computer program, (4) improving data of HA, (5) investigating the impact of a rolling planning horizon on the model, (6) limit the impact of the bottleneck operations and (7) investigate the use of the MIP model at order acceptance. A major implementation challenge is the integration with existing systems. The MIP model in this research is developed in AIMMS, a high-cost software. Possibilities for HA are acquiring an AIMMS license, designing the model in free software using Python or integrating the model directly into Isah. We recommend HA to discuss the possibilities with its ERP consultant and make a trade-off between user-friendliness, implementation time and implementation cost. Another challenge is the resistance of employees performing the planning for almost 20 years. We recommend using clear communication about the benefits of the model, addressing the concern of the stakeholders and involving the stakeholders in the implementation process.

Contribution to theory and practice

This research contributes to theory by introducing a novel resource loading MIP model that integrates both resource-driven and time-driven approaches, addressing constraints such as non-preemption, capacity, and precedence simultaneously. Literature on resource loading is limited, particularly for combining these two approaches. Additionally, this research offers a comprehensive description of the job shop production environment different from traditional job shop planning and scheduling problems. This environment serves as a foundation for researching similar production environments. The proposed MIP model and its components hold the potential for inspiring

future research in resource loading, guiding researchers in formulating similar problems with non-preemption constraints. This research contributes to practice by introducing a resource loading method for integrating resource capacity into HA's tactical production planning, enabling effective tactical decision-making. Employing the proposed MIP model for up to 2-month planning horizons allows HA to optimise production, providing insights into tardy orders and cost-efficient overtime allocation. The model also provides insights into the bottleneck operations at HA. Additionally, HA's potential to improve Customer Service Level beyond 90% through overtime usage is demonstrated.

Acknowledgements

Dear reader,

I would like to present you with my Master's thesis. This Master's thesis is written to finalise my Master's in Industrial Engineering and Management with a specialisation in Production & Logistics at the University of Twente.

First, I would like to thank Hortec Assemblies BV for providing an interesting problem at their company and their trust in me. I would like to thank Lars Zwanenburg for being my company supervisor and for providing me with valuable information whenever I asked for it. I would also thank the other colleagues, especially the operations manager and THT manager for providing me with information about their way of working.

Secondly, I would like to thank Engin Topan for being my first supervisor. He always provided me with valuable feedback and interesting questions to keep me sharp. He was always available for questions whenever I needed help during my Master's thesis. I would also like to thank Marco Schutten for being my second supervisor. He also provided valuable feedback during the process. He especially helped with difficulties designing the MIP model.

Last, I would thank my family, friends and boyfriend for showing their interest in my research and their support during the execution of this Master's thesis.

This Master's thesis is also the end of my student life. I had a great time at the University of Twente and learned a lot academically but also personally. I got great opportunities such as publishing my first paper during the course Purchasing Management together with two other Master students: Which strategies and corresponding competences are needed to improve supply chain resilience: A COVID-19 based review. Now it is finally time to start my working career.

I hope you enjoy reading it!

Mirthe Zwanenburg
Ootmarsum, August 2023

Contents

1 Introduction **1**
1.1 Company introduction - Hortec Assemblies B.V. 1
1.2 Research motivation 2
1.3 Problem context 3
1.3.1 Problems 4
1.3.2 Possible core problems 5
1.3.3 Core problem of HA 6
1.4 Scope 6
1.5 Research objective and questions 7

2 Current Situation **9**
2.1 Production environment at HA 9
2.1.1 Operations at SMT production 9
2.1.2 Operations at THT production 10
2.1.3 Occurrences of SMT and THT operations 11
2.1.4 Different product routings within HA 12
2.2 Current production planning method of HA 14
2.3 Planning functionalities in Isah 15
2.3.1 Product data 15
2.3.2 Operation planning parameters 16
2.3.3 Other planning functionalities within Isah 18
2.4 Performance of current due date setting and operational planning method of HA . . 19
2.4.1 Customer Service Level 19
2.4.2 Tardiness and earliness 19
2.5 Conclusion on the current situation 20

3 Literature review **22**
3.1 Resource loading 22
3.2 Solution approaches for resource loading 23
3.2.1 Constructive heuristic 23
3.2.2 Improvement heuristics 25
3.3 Conclusion on the literature study 27

4 Solution design **28**
4.1 MIP model formulation 28
4.1.1 Sets, parameters and decision variables 28
4.1.2 Objective function and constraints 30
4.2 Constructive heuristic 31
4.2.1 Select order based on priority rule 32
4.2.2 Plan orders 32
4.2.3 Calculate objective value 34
4.3 Improvement heuristic: Simulated Annealing 35
4.3.1 Neighborhood operator 35

4.4 Conclusion on solution design 37

5 Solution test **38**

5.1 Solution test setup 38

5.1.1 Experiments and their goals 38

5.1.2 Parameter settings 39

5.1.3 Data 39

5.2 Experiments on MIP 40

5.2.1 MIP: impact of planning horizon length & workload 40

5.2.2 MIP: impact of different maximum overtime 42

5.2.3 MIP: impact of different ϵ 43

5.2.4 MIP: insight in Customer Service Level 44

5.2.5 MIP: validation of infeasible production plans 45

5.3 Experiment on the constructive heuristic: priority rules 46

5.4 Experiments on improvement heuristic: initialisation parameters 47

5.4.1 Initial temperature 48

5.4.2 Markov Chain length, cooling scheme and stopping condition 49

5.5 Solution quality of constructive and improvement heuristic 50

5.6 Conclusion on solution test 51

6 Conclusion and recommendation **53**

6.1 Conclusion 53

6.2 Recommendations 54

6.3 Contribution to theory and practice 55

6.4 Implementation challenges 56

Appendices **60**

A Solution design: technical details constructive algorithm **60**

List of Figures

1.1 Graphical representation of PCBs through the production process. 1
1.2 Problem cluster of HA. 4
1.3 Positioning framework (from: Hans (2001)) 7
2.1 Graphical representation of the production methods at HA with their corresponding inputs and outputs. 9
2.2 Graphical representation of the operations flow at SMT production. 9
2.3 The SMT line at HA. 10
2.4 Wave soldering as represented in Arunasalam et al. (2022). 10
2.5 Flowchart of the current production planning method of HA. 14
2.6 Flowchart of creating an SMT and THT PO. 15
2.7 Example of an operation defined in Isah. 16
2.8 Example of priorities used within Isah. 19
3.1 Example of linear precedence relations as provided by Hans (2001). 22
3.2 Example of generalised precedence relations as provided by Hans (2001). 23
3.3 Graphical representation of FFL adjusted from Thurer et al. (2014). 24
3.4 Graphical representation of PFFL adjusted from Robinson and Moses (2006). 25
3.5 Illustration of local and global optimality by Jeon et al. (2017). 26
4.1 Flowchart of the constructive heuristic. 32
4.2 Flowchart of step 2 (plan orders) in the constructive heuristic. 33
4.3 Flowchart of step 2.2. (plan (i,s) in (consecutive) time bucket(s)) of the constructive heuristic. 34
4.4 Flowchart of the neighborhood operator. 37
5.1 Distribution of processing times in dataset of 2023. 40
5.2 Results initial temperature experiment with Markov Chain length of 100. 48
5.3 Results initial temperature experiment with Markov Chain length of 1000. 48
5.4 Results of the Markov Chain length and cooling scheme experiments. 49

List of Tables

1.1 Customer Service Levels of HA from 2017-2021. 2
1.2 Revenue growth of HA from 2017-2023. 3
1.3 % of deliveries arrived too late at HA in 2017-2022. 6
2.1 Occurrences of operation at SMT production. 12
2.2 Occurrences of operation at THT production. 12
2.3 Ten most occurring product routings at HA. 13
2.4 Distribution of production routes outside the top ten at HA. 13
2.5 Rules used to estimate lead time per operation on product level. 16
2.6 Settings used within HA for the parameters per operation in Isah. 18
2.7 Percentages of early POs, POs on-time and tardy POs of HA. 20
4.1 Overview of the operations at HA in the set J_j. 28
4.2 Sets, parameters and decision variables 29
4.3 Example of auxiliary variables non-preemption constraints. 30
5.1 Mean processing time per operation type j in 2023. 39
5.2 Definition of workload types at HA. 41
5.3 Scenarios test MIP: impact of planning horizon length & workload. 41
5.4 Result test MIP: impact of planning horizon length & workload. 42
5.5 Result test MIP: impact of maximum overtime. 43
5.6 Scenarios test MIP: impact of different ϵ. 43
5.7 Result test MIP: impact of different ϵ. 44
5.8 Result test MIP: Customer Service Level. 45
5.9 Result test MIP: validation on infeasible production plans. 46
5.10 Performance of constructive heuristic for different priority rules: what maximum overtime needed? 47
5.11 Performance of constructive heuristic for different priority rules using same amount of maximum overtime. 47
5.12 Impact of α on the number of iterations needed for SA. 49
5.13 Performance of the three models: tardy orders and objective values. 50
5.14 Performance of the three models: the gap between the solutions. 51

Acryonyms

AOI Automated Optical Inspection
BFL Backward Finite Loading
FFL Forward Finite Loading
HA Hortec Assemblies BV
Isah The ERP-system used at HA
KPI Key Performance Indicator
MTO Make-To-Order
PCB Printed Circuit Board
PO Production order
SA Simulated Annealing
SMT Surface-Mount Technology
TA Tabu Search
THT Through-Hole Technology

1 Introduction

This chapter discusses the research problem and research plan. Section 1.1 introduces the company of this research, Hortec Assemblies BV. Section 1.2 elaborates on the research motivation and Section 1.3 clarifies the problem context. Section 1.4 defines the scope. Section 1.5 defines the research objective and states the research questions and approach.

1.1 Company introduction - Hortec Assemblies B.V.

Hortec Electronics was founded in 1998. Hortec Electronics specialised in the development and assembly of electronics. Since January 2021, the engineering and assembly departments are divided into Hortec Technology BV and Hortec Assemblies BV, respectively. This research is executed in the management department at Hortec Assemblies BV (hereinafter referred to as HA). HA produces and assembles (control) electronics for third parties that manufacture for aerospace, railway, automotive and industry. The key focus of HA is passion, flexibility and quality. HA uses a Make-To-Order (MTO) production strategy. With an MTO strategy, products are made in response to placed customer orders such that they can be highly tailored to customers' needs (Peeters and van Ooijen, 2020). An MTO strategy is characterised by long lead times, low storage costs and high flexibility (Peeters and van Ooijen, 2020).

This Figure is not publicly available due to confidentiality.

Figure 1.1: Graphical representation of PCBs through the production process.

Figure 1.1 shows the stages of the Printed Circuit Bord (hereinafter referred to as PCB) before and after the different production methods at HA. The production starts with an empty PCB, represented left in Figure 1.1. The empty PCB is an input for the first production method at HA called Surface-Mount Technology (SMT). With SMT, electrical components are mounted directly on the surface of a PCB. Commonly used SMT components are chip resistors, metal electrode face resistors, chip capacitors, chip inductors, discrete semiconductors and integrated circuits (Lee, 2001). The output of the SMT production method is called a semi-finished product and is represented in the middle of Figure 1.1. The output of the SMT production method is the input for the second method called Through-Hole Technology (THT). With THT production, larger electrical components that are not suitable for SMT are placed on PCBs through a hole and are mounted with soldering tin that passes through the hole (Vianco and Feng, 2016). Commonly used THT components are inductor coils, relays, connectors, switches, and fuse holders (Vianco and Feng, 2016). The output of the THT production method is called the end product and is represented right in Figure 1.1. Every product at HA follows these stages.

1.2 Research motivation

Due date setting and production planning need to be effective and efficient due to the increase in customers' demand for short lead times. Both are performed manually at HA with the help of data and planning functionalities in the ERP system called Isah. Three main motives for this research are identified:

Unreliable and low Customer Service Level. The most essential motive of this research is the unreliable and low Customer Service Level at HA. A Customer Service Level is the fraction of customer orders filled on or before their due dates (Sawik, 2006). HA knows three types of due dates:

- **The preferred due date of the customer.** Suggested by the customer but not achievable since this is not based on the internal lead times of HA and customers want products as fast as possible. Therefore, the preferred customer due date is currently not taken into account for the key performance indicator (hereinafter referred to as KPI) Customer Service Level.
- **The first confirmed due date.** Agreed during the customer ordering process. The first confirmed due date is created by HA based on the current production plan and available resource capacity.
- **The last confirmed due date.** Agreed during the purchase or production process. It happens that the first confirmed due date cannot be achieved due to disruptions during the purchase or production process. If the delay turns out to be more than one week, the first confirmed due date is changed in accordance with the customer to a new due date called the last confirmed due date. In the remainder of this research, we refer to the last confirmed due date as the external due date.

The Customer Service Level of HA is measured by calculating the deviation between the due date and the actual shipment day of a product. Table 1.1 shows the Customer Service Level of HA from 2017 until 2021 based on the first confirmed due date and the last confirmed due date.

Table 1.1: Customer Service Levels of HA from 2017-2021.

	2017	2018	2019	2020	2021
First confirmed due date	64.5%	74.6%	67.5%	74.4%	79.6%
Last confirmed due date	64.8%	85.4%	82.7%	83.6%	86.6%

Currently, HA focuses on the Customer Service Level based on the last confirmed due date since the customer agreed with this new due date. However, it would be ideal to measure the Customer Service Level based on the first confirmed due date since this date is initially agreed upon. However, improving the first confirmed due date from 79.6% to 90% is a huge step. This is the future goal of HA. From 2017 to 2018, the last confirmed due date, which is the service level accepted at HA, increased and became stable. However, the Customer Service Level is still too low. The aim of management is to achieve a Customer Service Level based on the last confirmed due date of at least 98%. According to management, the Customer Service Level aim of 98% is not achievable by only improving the production planning method. Therefore, management aims to achieve a Customer Service Level based on the last confirmed due date of at least 90% by conducting this research. Since the Customer Service Level is the most important motive, the action problem of this research is:

The Customer Service Level based on the last confirmed due date of HA needs to be increased from 86.6% to at least 90%.

Recommendation of earlier conducted Master Thesis. Another motive is a Master Thesis conducted by Anouk Scholten at HA in 2020 to search for inventory reduction approaches (Scholten, 2020). One recommendation of this Master Thesis is to improve production planning since planning decisions are based on the experience and gut feelings of employees. No operational and limited tactical planning is used within HA (Scholten, 2020).

Company growth. The last motive is the aim of company growth. Currently, HA is growing and is planning to grow in the upcoming years. Production planning based on employee experience and gut feelings is feasible but not efficient. When HA grows it will no longer be feasible to plan production manually. HA wants to be prepared for this stage. Table 1.2 shows the revenue growth of HA over the years between 2017 and 2023. The revenue growth of 2022 is based on the revenue until October plus a forecast for November and December. The revenue growth of 2023 is a combination of already placed orders and a forecast. Since 2019 HA is growing, except for 2020 which is due to COVID-19.

Table 1.2: Revenue growth of HA from 2017-2023.

	2017	2018	2019	2020	2021	2022	2023
Revenue (x €1.000.000)	3,04	2,67	3,76	2,95	3,59	5,62	6,20
Revenue growth compared to year before (%)	-6	-12	41	-21	22	56	10
Cumulative revenue growth since 2016 (%)	-6	-18	22	1	22	79	89

1.3 Problem context

To identify the core problem of the action problem of HA, a problem cluster is used to map all problems along with their connections (Heerkens and van Winden, 2017). Figure 1.2 shows the problem cluster of the action problem of HA. This problem cluster is identified based on the experience of the director and SMT production manager. Section 1.3.1 explains all problems (white boxes) within the problem cluster, Section 1.3.2 explains all possible core problems (grey boxes) and Section 1.3.3 explains the chosen core problem (blue box).

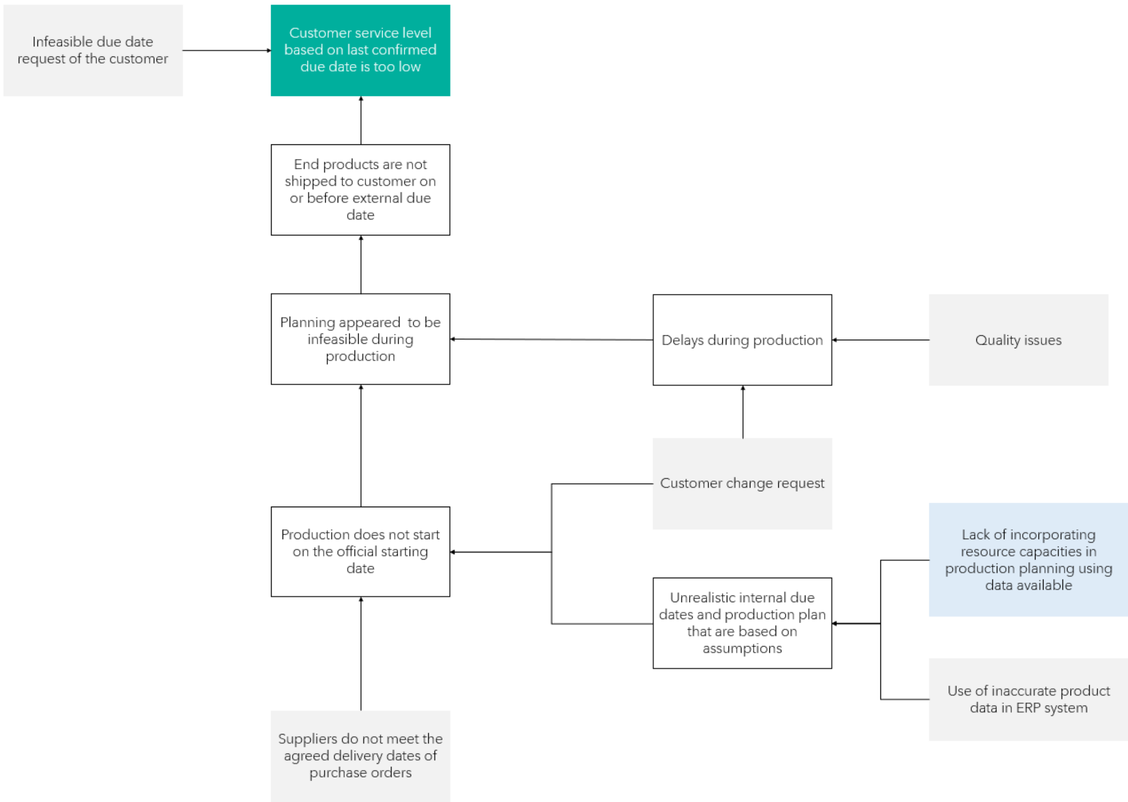

Figure 1.2: Problem cluster of HA.

1.3.1 Problems

End product is not shipped on or before the external due date. No new external due date is negotiated with the customer for a shipment delay of less than one week. HA ships the product too late to the customer resulting in a lower Customer Service Level. For a shipment delay of at least one week, a new external due date is negotiated with the customer: the last confirmed due date. Table 1.1 shows the Customer Service Level of HA based on production orders (hereinafter referred to as POs) shipped on time. The other part is shipped too late.

Delays during production. A delay during production can be any activity that increases the flow time of a PO. The flow time of a PO is the amount of time a PO spends in the production process (i.e. total processing time). The production end date (hereinafter referred to as the internal due date) is not achieved when delays during production occur since the planning does not include a margin for production delay. Section 1.3.2 explains the two main reasons for production delay: quality issues and customer change requests.

Production does not start on the official starting date. It happens that the production of a PO does not start on the official start date. If that happens the planning becomes infeasible since there are no margins for a later start of production. Due to no margins, the internal due date delays and the product is shipped too late. The starting day of the next POs also delays due to tight planning without margins.

Unrealistic internal due dates and production plans based on assumptions. One reason for delays in the production start is unrealistic internal due dates and production planning based on assumptions. After receiving a purchase order from a customer the operations manager sets an external due date and plans a PO as soon as possible based on available resource capacity. He identifies the available resource capacity by his experience and the production plan so far. The external due date and internal due date are set by the operations manager and Isah calculates the start date based on the production step lead times that are entered in Isah by the job preparator. These production step lead times are assumed based on experience. In reality, these production step lead times are often higher than pre-calculated. Isah also does not take into account the days off and vacations of employees. So more resource capacity is taken into account than in practice available which results in a period between the start date and internal due date of production that is too small.

1.3.2 Possible core problems

Infeasible due date request of the customer. Section 1.2 states that the due date preferred by the customer is (almost) always infeasible since the customer has no knowledge about the internal lead times of HA. The infeasible customer due date requests are not taken into account to calculate the Customer Service Level so solving this problem will not increase the Customer Service Level. Therefore, this is not our core problem. Improving the preferred customer due date is outside the scope of this research.

Quality issues. Production can be delayed due to quality issues. PCBs need rework when they do not meet the quality standards of HA. PCB rework happens monthly. The production employees are receiving yearly training to limit the chance of PCB rework. Therefore, this is not the chosen core problem.

Customer change request. A customer of HA can send a request for change anywhere in the process. A change request can be the adjustment of one or more components, the adjustment of a process step, or the adjustment of the required number of products. The customer can even cancel the PO. Due to a customer change request production is put on hold and delayed. Customer change requests also lead to a higher inventory position when components are already purchased and in stock. Customer change requests are not our core problem since HA wants to have certain flexibility for its customers.

Suppliers do not meet the agreed delivery dates. Production cannot start and is postponed when a component is not delivered on or before the production start date. The purchase department already uses a buffer of two weeks between the delivery of the components and the start of production to ensure production start dates are achievable. To decrease inventory HA aims at a buffer of at most 3 days. This implies that a component may arrive 3 days before or after the agreed delivery date. Both arriving too early and too late are unacceptable since arriving too early unnecessarily increases the inventory position of HA. Deliveries too late lead to delayed production. For this research only the orders too late are important. Table 1.3 gives an overview of the purchase orders delivered too late by the suppliers of HA.

Table 1.3: % of deliveries arrived too late at HA in 2017-2022.

	2017	2018	2019	2020	2021	2022
Deliveries too late > 3 days (%)	7.5%	7.9%	8.9%	7.3%	9.0%	8.2%

The main reason for unmet delivery dates is component scarcity. COVID-19 created a component scarcity in the electronics industry. Components could not be produced and shipped to Europe due to lockdowns in Asia. The COVID-19 situation became stable but the component scarcity is not solved yet. Specific components are not available or are delivered later. Purchase orders delivered months to years later is normal. To deal with postponed deliveries substitute components are purchased, the PCB is redesigned or production is postponed. These steps lead to a delayed production start. This is not our core problem since we cannot influence the suppliers and the component scarcity.

1.3.3 Core problem of HA

The last possible core problem is the lack of incorporating resource capacities in production planning. Section 1.3.1 explained that assumptions are used for internal due date setting and production planning. One assumption is made on production step lead times that are entered in Isah. The other assumption is the available resource capacity taken into account at internal due date setting and production planning. We choose to solve the problem of using available resource capacity in production planning. Even if HA enters the right production step lead times in Isah the internal due dates and production plans are still too realistic since available resource capacity based on data available is not considered. By providing a solution to using available resource capacity in tactical production planning the Customer Service Level of HA improves. By achieving internal due dates, external due dates can also be achieved which leads to the improvement of the Customer Service Level. HA can improve production step lead times by itself to improve the Customer Service Level even more. Since this core problem will solve other problems within the problem cluster and create efficient production planning that leads to an increase in Customer Service Level, the chosen core problem of HA is:

Lack of incorporating resource capacities in production planning using data available within HA.

1.4 Scope

The core problem of HA is production planning. HA does not distinguish between the three different levels of planning: strategic, tactical and operational. The operations manager plans the PO in Isah based on the external due date and the production employees try to follow the schedule by starting the next PO based on the earliest starting date or the earliest internal due date. Section 2.2 discusses the current production planning method of HA in more detail. To limit the width of this research, we identify which type of production planning level is most relevant to this problem. Hans (2001) provides a positioning framework to distinguish between different capacity planning functions. Figure 1.3 shows the positioning framework.

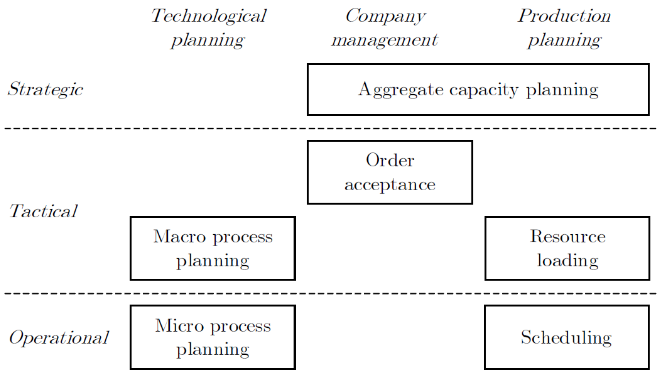

Figure 1.3: Positioning framework (from: Hans (2001))

Hans (2001) explains the different production planning levels as follows. Strategic production planning involves long-range decisions made by management such as building new facilities or buying new machines. Tactical planning involves medium-range decisions such as allocating sufficient resources to deal with incoming demand as effectively and profitably as possible. Tactical planning is divided into two activities: order acceptance and resource loading. Order acceptance is about accepting or rejecting incoming orders based on the order characteristics and the current state of the production system. Resource loading is about loading a given set of orders and determining reliable internal due dates and the resource capacity levels needed to process these orders and their constituting jobs. Resource loading can establish the feasibility and suitability of a given set of accepted orders. Operational planning concerns the short-term scheduling of production orders resulting from the tactical level. The result of operational planning is the assignment and sequence of the jobs on machines.

For HA, we create a method that incorporates resource capacity in tactical production planning such that the last confirmed due date is achieved and the Customer Service Level increases. It is not important to which specific machine a job is assigned but to which time bucket and against which resource capacity levels. In other words, we plan a given set of orders and need to determine internal due dates and resource capacity levels such that the Customer Service Level increases. Therefore, we focus on the tactical production planning activity resource loading.

1.5 Research objective and questions

The goal of this research is to increase the Customer Service Level from 86.6% to at least 90%. The main objective of this research is to design a method for a high-mix low-volume MTO company that incorporates resource capacity into tactical production planning using data available while improving the Customer Service Level. The research objective leads to the following main research question:

How can a method that takes into account resource capacity using data available in tactical production planning for a high-mix low-volume make-to-order EMS company be designed such that the Customer Service Level improves from 86.6% to 90%?

To answer the main research question, several sub-questions are determined.

1. *What does the current production planning of HA look like?*

   To incorporate resource capacity in a production tactical planning method for HA, the current production planning of HA should be clear. This research question is answered in Chapter 2 by the following sub-questions:

   (a) What does the production environment of HA look like?

   (b) What does the production planning method of HA look like?

   (c) What are the (tactical) planning functionalities in Isah?

   (d) What is the performance of the current production planning method in terms of Customer Service Level, tardiness and earliness?

2. *What literature is available on designing and solving a resource loading method that is effective for a high-mix make-to-order job-shop type production process*

   After analysing the current situation, a literature study is executed in Chapter 3 to gain information on resource loading. To answer this research question several sub-questions are determined:

   (a) What is resource loading and how can it be used in tactical production planning?
   (b) Which modelling approaches are used for solving resource loading?

3. *How can a tactical planning method incorporating resource capacities for HA be designed?*

   After analysing the literature, the tactical planning method for HA needs to be designed. Chapter 4 shows the solution design by answering the following sub-research questions:

   (a) What does the exact model formulation of the resource loading method for HA look like?
   (b) What does the constructive heuristic for the resource loading method for HA look like?
   (c) What does the improvement heuristic for the resource loading method for HA look like?

4. *What is the effect and improvement of the proposed method for HA?*

   The proposed solution in Chapter 4 is tested on its effect and improvement in Chapter 5. To answer the research question several sub-questions are determined:

   (a) Which data is used to test the designed models?

   (b) What is the performance of the designed models?

   (c) To what extent can the exact model help to improve the Customer Service Level?

Chapter 6 provides the conclusion of this research, recommendations for HA and recommendations for future research. It also discusses the contribution of this research to theory and practice.

2 Current Situation

This chapter analyses the current situation at HA. Section 2.1 describes the production environment at HA. Section 2.2 describes the current production planning method at HA. Section 2.3 describes the planning functionalities in Isah. The performance of the current planning method of HA is investigated in section 2.4. Section 2.5 provides a conclusion on the current situation analysis.

2.1 Production environment at HA

Section 1.1 describes that HA knows two production methods: SMT and THT. Figure 2.1 shows a graphical representation of the two production methods with their main inputs and output.

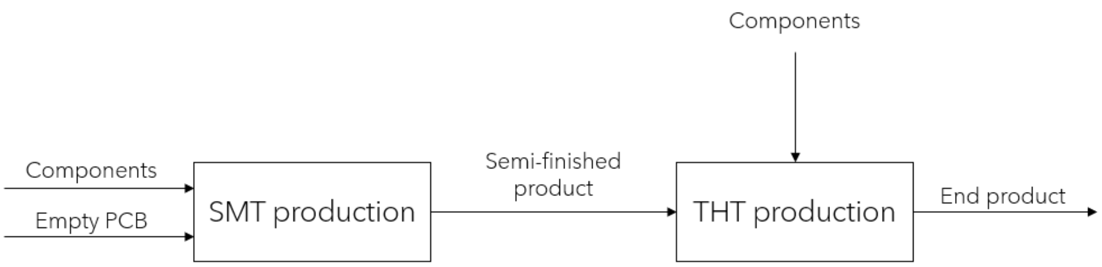

Figure 2.1: Graphical representation of the production methods at HA with their corresponding inputs and outputs.

Both production methods exist out of several operations. Section 2.1.1 describes the operations of the SMT department and Section 2.1.2 describes the operations of the THT department. Section 2.1.3 analysis the occurrences of each operation in the SMT and THT department. Section 2.1.4 analysis the different production routes at HA.

2.1.1 Operations at SMT production

Figure 2.2 shows the flow of the operations at SMT production. In practice, all products follow this flow but not all product routings are inserted correctly in Isah. Therefore, the product routings in Isah may deviate from the flow in Figure 2.2.

Figure 2.2: Graphical representation of the operations flow at SMT production.

The following operations are represented:

- **SMT.** Figure 2.3 shows the SMT line with a stencil printer, two SMT pick & place machines and the SMT oven. First, A PCB moves through the stencil printer which places paste on the PCB. Then the PCB moves through the two SMT pick & place machines that place the components on the PCB and eventually, it will move through an oven that adheres the components to the PCB.

Figure 2.3: The SMT line at HA.

- **AOI.** AOI stands for Automated Optical Inspection and is used to check the quality of the SMT component placements and paste.
- **Rework.** When a fault is detected during AOI, the PCB needs rework to repair the fault.

2.1.2 Operations at THT production

For THT production no general flow of operations exists. The following operations can be performed at THT production:

- **Conventional.** Conventional includes cutting the PCB panel into individual PCBs, preparing the components, and placing the components on the PCB.
- **Wave soldering.** A few years ago, HA only used wave soldering. Figure 2.4 shows the general idea of wave soldering. First, the PCB moves through a fluxing station which cleans the components to be soldered. Next, the PCB moves through a preheat zone and eventually passes through the soldering station (Vianco and Feng, 2016). Lastly, the PCB moves through a cooling station. In practice, wave soldering is not used anymore but not all product routings are updated in Isah. Therefore some product routes still include wave soldering.

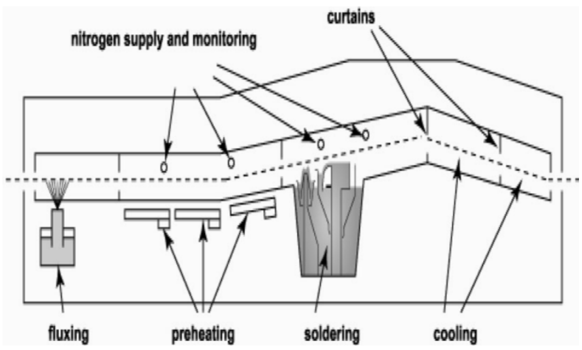

Figure 2.4: Wave soldering as represented in Arunasalam et al. (2022).

- **Selective soldering.** For three years HA has had a selective soldering machine. With selective soldering a soldering head solders the through-hole joints individually instead of moving the PCB through a solder wave. Whereas wave soldering is faster, selective soldering uses less materials-solder, flux and power. Besides, selective soldering results in less rework and less post-assembly cleaning.

- **Post-assembly.** After wave or selective soldering, a PCB may need post-assembly. This includes removing flux residues, checking the soldering joints, removing solder balls, manual soldering et cetera.

- **PCB washing.** PCBs that need coating and PCBs with sensitive and expensive modules used in aerospace, automotive, medical technology and telecommunications need to be washed. The main task of washing is to remove flux residues, oxides and soldering materials (kolb Cleaning Technology GmbH, n.d.).

- **PCB drying.** PCBs that are washed also need to dry.

- **Potting.** With potting, the PCB is placed in a mould and a liquid compound is poured into the mould. The liquid compound solidifies and the PCB is encased by the compound (MPE-electronics, b).

- **Coating.** With coating, the PCB is covered by a polymeric film to protect the PCB from environmental and physical factors (MPE-electronics, a).

- **Programming.** Some PCBs need to be programmed with specific software. This software is delivered by the customer or by the engineering department of Hortec Technology.

- **Test & Repair.** Some PCBs need to be tested. If a PCB fails a test, the reason for failing is discovered and the PCB is repaired such that it will pass the test eventually.

- **Burn-in test.** With a burn-in test the PCB is tested on the customer- and product-specific requirements for 12, 24 or even more hours.

- **Intermediate check.** Sometimes an internal check is needed between production steps to ensure that the product still meets the quality standard of HA.

- **Final check.** All PCBs get a final check before packing to ensure that the quality threshold of HA is achieved.

- **Packaging.** Packaging includes packing the products for shipment.

2.1.3 Occurrences of SMT and THT operations

To indicate the occurrence of an operation, the semi-finished and end products in Isah updated in 2017 or later are analysed. In total 757 semi-finished products and 950 end products are analysed. Table 2.1 shows the occurrences of the operations of SMT production based on the 757 analysed semi-finished products.

Table 2.1: Occurrences of operation at SMT production.

Operation	Occurrence
SMT	690
AOI	538
Rework	497

Table 2.2 shows the occurrences of the operations of THT production based on the 950 analysed end products.

Table 2.2: Occurrences of operation at THT production.

Operation	Occurrence
Conventional	1695
Final check	855
Packaging	751
Test & repair	471
Selective soldering	347
Intermediate check	321
Wave soldering	187
Coating	87
Potting	76
PCB Washing	74
Burn-in test	40
Post-assembly	26
Program	15
PCB Drying	8

From the operation occurrence analysis, we can conclude that the product routings in Isah contain inaccuracies. Two main points that motivate the inaccuracies:

- Section 2.1.2 explained that HA does not use its wave soldering machine anymore. The occurrence of wave soldering is still 187 whereas it should equal zero.
- Section 2.1.1 explained that SMT production always follows the flow in Figure 2.2. This implies that the occurrence of SMT, AOI and rework should equal 757 each.

These two points do not influence the result of this research since the solution design is tested with the product routings from Isah and not with improved product routings. However, correctly designed product routings improve the Customer Service Level since lead times are better estimated when article routings are inserted correctly. Updating the product routings is outside the scope of this research but is a recommendation for future research.

2.1.4 Different product routings within HA

The previous section determines the occurrences of individual operations. This section analyses the different product routings at HA. After analysing the product data for products updated in 2017 or later, a total of 313 different product routings are identified over 1707 products. Table 2.3 shows the 10 most occurring product routings, these routings cover 57% of all product routings at HA.

Table 2.3: Ten most occurring product routings at HA.

Route	**Occurrence (products)**	**Occurence (%)**
SMD, AOI, REW	256	15
WSMD,SMD,WAOI,AOI,REW	179	10
SMD	139	8
PR,SSOL,PR,EC,VER	101	6
PR,EC,VER	71	4
SMD,AOI	60	4
PR,SOL,PR,EC,VER	52	3
PR	51	3
PR,SSOL,EC,VER	35	2
SMD,REW	31	2
Total	**975**	**57**

The product routings also prove that inaccuracies in Isah exist. One main motivation is that there are several combinations of SMT, AOI and rework whereas in practice there is only one route possible: SMT, AOI and rework. Table 2.4 shows the distribution of the product routing occurrences for the routings outside the top ten.

Table 2.4: Distribution of production routes outside the top ten at HA.

Occurence	**# of production routes**
1	172
2	61
3	22
4	12
5	9
6	4
7	8
8	2
9	1
10	3
12	2
13	1
14	1
16	1
17	1
18	1
20	1
22	1

The SMT department is a flow shop since every product follows the same route: SMT, AOI and Rework (Pinedo, 2016b). The THT department is a job shop since every product follows its own predetermined route (Pinedo, 2016b). There are recurring product routes at the THT department but to keep it simple the SMT and THT production environments are analysed as a job shop.

2.2 Current production planning method of HA

Figure 2.5 shows the current production planning method of HA. An incoming sales order from a customer triggers the process. The next paragraphs describe the different tasks of the flowchart.

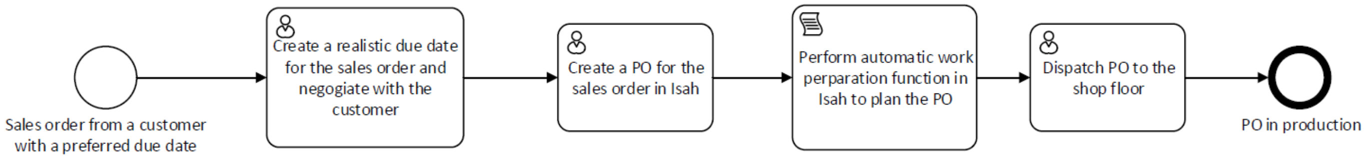

Figure 2.5: Flowchart of the current production planning method of HA.

Incoming sales order from a customer with a preferred due date. The process is triggered every time a sales order arrives. The sales order includes the following information: product, quantity and preferred due date. Sales orders arrive at random.

Create a realistic due date for the sales order and negotiate with the customer. Section 1.3.2 describes that the due date request of a customer is (almost) always infeasible. The operations manager of HA creates a feasible external due date for the PO based on the customer's preferred due date, the current production plan, his experience, the availability of critical components and the urgency of the sales order in the following way:

1. He checks the current production plan and selects a week as close as possible to the customer-preferred due date in which enough resource capacity is left. The amount of resource capacity left in a week is based on the experience and the gut feeling of the operations manager.
2. He checks if critical components with long lead times can be delivered before or in the selected week. If all critical components can be delivered on time, the first possible date in the selected week is set as the external due date. If components cannot be delivered on time, a later external due date is chosen.
3. For a high-urgency sales order, an earlier external due date may be chosen but this leads to changes in the overall production plan.

The external due date is negotiated with the customer and added to the sales order in Isah as the first confirmed due date. Almost all sales orders at HA are accepted. Manual due date setting has a major disadvantage. It does not ensure the earliest feasible starting date possible based on available resource capacity in Isah, which leads to higher customer lead times and infeasible due dates. Therefore, it would be better to create a due date based on available resource capacity in Isah such that the earliest possible start date and internal due date are selected.

Create POs for the sales order in Isah. The second step is to create a PO for the sales order in Isah. For both SMT and THT production, a separate PO is created. Due to differences in batch quantities and the way Isah is designed, it is best to use one PO for the SMT department and one PO for the THT department. This is not changed during this research. The operations manager gives an internal due date to the two types of POs. For THT POs the internal due date equals the external due date minus two slack days and for SMT POs the internal due date is 7 days before the start of the THT PO. Figure 2.6 graphically represents this process.

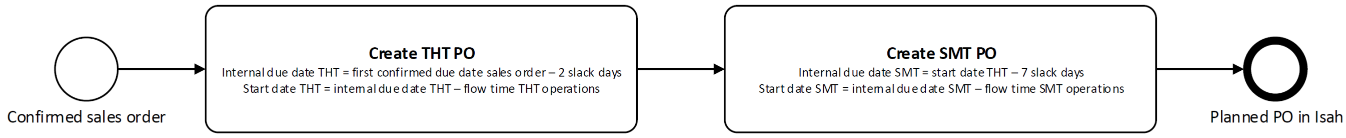

Figure 2.6: Flowchart of creating an SMT and THT PO.

Perform automatic work preparation function in Isah to plan PO. The third step is performing automatic work preparation in Isah to create a start date for a PO. Automatic work preparation is executed per PO. With automatic work preparation, Isah automatically executes the following steps:

1. **Copying product data to PO.** Product data includes the product routings, the components, and the tools needed to produce the product.
2. **Plan the PO by creating a start date with backward planning.** This means that starting from the internal due date all production steps are scheduled backwards to a starting date (Yeh, 2000). Section 2.3 explains the planning functionalities in Isah in depth.

Dispatch PO to the shop floor. The SMT department dispatches the POs to the shop floor based on the earliest starting date. The THT department dispatches the POs to the shop floor based on the earliest external due date since the THT manager does not trust the production plan in Isah. The management of HA aims at dispatching based on the earliest starting date of the POs since these are created based on internal and external due dates.

2.3 Planning functionalities in Isah

Section 2.2 states that Isah is used to plan POs based on backward planning. Isah needs the following data to plan a PO: product data (described in Section 2.3.1) and operation planning parameters (described in Section 2.3.2). Isah also has more planning functionalities than backward planning. Section 2.3.3 explains the planning functionalities unexploited at HA to give an impression of the scope of Isah.

2.3.1 Product data

The automatic work preparation function copies the product data to the PO. This includes the product routings, and the components and tools needed to start production. Each operation in the production route of a product has a lead time (in minutes) defined to produce one PCB. The lead time per operation for a specific product is estimated by the work preparator with a calculation sheet and his experience. Table 2.5 shows the lead times (in seconds) that are used in the calculation sheet for the lead time estimation of the operations of conventional, selective soldering and manual soldering on product-level. The total lead time of the conventional operation is calculated with a calculation sheet by counting the number of components to assemble and multiplying it with the lead time of that type of component. For selective and manual soldering an estimation of 3.5 seconds and 7 seconds per soldering of one through-hole joint is used. The lead time in seconds is converted to a lead time in minutes. This information is not regularly controlled and updated and therefore may be outdated. For all other operations, estimation is based on the experience of the THT manager and work preparator.

Table 2.5: Rules used to estimate lead time per operation on product level.

Operation	**Description**	**Lead time (seconds)**
Conventional	Placement 2 pins component curved	20
Conventional	Placement 2 pins component	20
Conventional	Placement 3 pins component	20
Conventional	Placement IC	25
Conventional	Placement connector	35
Conventional	Placement of trafo	80
Conventional	Placement of relais	40
Conventional	Placement resistor network	20
Selective soldering	Soldering of one through-hole joint	3.5
Manual soldering	Solder of one through-hole joint	7

2.3.2 Operation planning parameters

Parameters are defined manually per operation (in Dutch: *Capaciteitsgroepen*). These operation parameters are used when planning a PO in Isah. The operation parameters in Isah are explained to get insight into the planning functionalities in Isah. Figure 2.7 shows an example of an operation defined in Isah.

This Figure is not publicly available due to confidentiality.

Figure 2.7: Example of an operation defined in Isah.

The following operation parameters are used within the planning of HA:

- **Based on (*in Dutch: Op basis van*)**: plan the operation based on *employee-hours* or *machine-hours*. Man-hours are used when specific employees need to be assigned to an operation. Machine hours are used when an operation needs to be assigned in general, the assignment of the operation is not important for the production plan and can be done on the production floor.
- **Planning type (*in Dutch: Type*)**: plan the operation based on *finite* or *infinite capacity.* With finite capacity, Isah takes into account already planned POs and available capacity. Isah ensures that capacity is not exceeded. Standard a First Come First Serve priority is used with finite capacity planning, but the user can assign other priorities such as due date or customer

priority. With infinite capacity, Isah plans the operations consecutively without taking into account capacity. The planner manually analysis if capacity is exceeded and if the PO has to be rescheduled. The planning type can be assigned to each operation individually. This means that if the conventional operation is planned with finite capacity, already planned POs are taken into account and the conventional operation of the PO to plan may not exceed capacity. If the conventional operation is planned with infinite capacity, the conventional operation of the PO to plan may exceed capacity.

- **Schedule till (*in Dutch: Inplannen tot*)**: percentage of the available capacity for the operation that may be scheduled to execute the operation. This percentage is calculated over the parameter "to schedule per day". For example, if this parameter is set to 90%, 90% of the maximum capacity per day for this operation may be scheduled. The other 10% is reserved for urgent sales orders.
- **Maximum capacity per day (*in Dutch: Te plannen per dag*)**: the hours of capacity available per day. For example, HA has three employees available with 8 working hours per employee. If the maximum capacity per day is set to 8, the PO is assigned to one employee and the other 2 employees can be assigned to other POs. However, if the maximum capacity per day is set to 24 the PO is assigned to the three employees and no employee is left to work on other POs.
- **Consecutively (*in Dutch: Aaneengesloten*)**: in literature, this is also called non-preemption (Pinedo, 2016b). This means that any intervening days (after the start day) must be unoccupied and that the remainder must fit on the last day. So a job once started at a machine may not be interrupted until it is finished. Planning nonconsecutive means that all available capacity of the operation can be used even if this means that the job is interrupted to process another job, in literature called preemption. The parameter consecutively can only be used when planning with finite capacity since planning with infinite capacity always leads to consecutive operations.
- **Minimal lead time (*in Dutch: Minimale doorlooptijd*)**: the minimum number of days that are planned for an operation. For example, a potted PCB needs to dry for at least one day. The minimal lead time of PCB potting is set to 1 day. Isah will plan the potting operation for at least one day independent of the product. The minimal lead time is indicated in days. A minimum lead time of zero means that Isah does not take into account a minimal lead time.
- **Fixed lead time (*in Dutch: Vaste doorlooptijd*)**: a fixed number of days that are planned for the operation. For example, PCB drying always needs 1 day independent of the product. The fixed lead time is indicated in days. A fixed lead time of zero means that Isah does not take into account a fixed lead time.

Table 2.6 shows the manually inserted settings in Isah used within HA for the parameters of the operations. One example to clarify the parameters: the operation SMT is planned based on machine hours while taking into account infinite capacity. This implies that capacity may exceed when needed and the scheduler needs to check if that happens. The maximum capacity per day is 8 hours, from which 90% may be scheduled. The operation needs to be planned consecutively, so preemption is not allowed. For SMT, minimum and fixed lead times are not taken into account.

Table 2.6: Settings used within HA for the parameters per operation in Isah.

Operation	Based on	Planning type	Schedule till (%)	Maximum capacity per day (hours)	Plan consecutively	Minimal lead time (days)	Fixed lead times (days)
SMT	Machine-hours	Infinite capacity	90%	8	Yes	0	0
AOI	Man-hours	Finite capacity	90%	8	No	0	0
Rework	Man-hours	Finite capacity	90%	8	No	0	0
Conventional	Man-hours	Finite capacity	90%	32	No	0	0
Wave soldering	Man-hours	Finite capacity	90%	8	No	0	0
Selective soldering	Man-hours	Finite capacity	90%	8	No	0	0
Post-assembly	Man-hours	Finite capacity	90%	32	No	0	0
PCB washing	Man-hours	Infinite capacity	90%	8	Yes	1	0
PCB drying	Machine-hours	Finite capacity	90%	8	No	0	1
Potting	Man-hours	Infinite capacity	90%	8	Yes	1	0
Coating	Man-hours	Infinite capacity	90%	8	Yes	1	0
Programming	Man-hours	Finite capacity	90%	24	No	0	0
Test & Repair	Man-hours	Infinite capacity	90%	8	No	1	1
Burn-in test	Man-hours	Infinite capacity	100%	0	Yes	2	2
Intermediate check	Man-hours	Infinite capacity	90%	8	Yes	0	0
Final check	Man-hours	Finite capacity	90%	8	No	0	0
Packaging	Man-hours	Infinite capacity	90%	8	Yes	0	0

Within HA, nobody is responsible for updating and controlling the parameters. No one knows the exact meaning of the parameters and their influence on the planning. This implies that backward planning within HA is not used to its fullest potential and is conducted with outdated parameter values that may not reflect reality. One example is the parameter setting for conventional, where 3 employees are available per day to work on the conventional operation of one PO. However, in practice, only one employee is working on the conventional step of one PO. The other remaining employees can work on other POs. Due to the parameter setting, 3 employees are planned on the conventional step of one product so more capacity is assumed than available in practice.

2.3.3 Other planning functionalities within Isah

The following functionalities are also available within Isah but are not used:

- **Plan based on production status/-date.** HA plans one PO at a time. However, it is also possible to plan all POs within a specific production status or a specific planning horizon. Within Isah, production statuses are used to keep track of all planned POs. For example, one status is "THT - complete and waiting to be released to the work floor".
- **Forward planning.** Forward planning plans the operations of a product from the scheduled starting date until all operations are performed (Yeh, 2000). One possible application of forward planning is to identify a suitable internal due date by checking if capacity is exceeded when we want to produce a PO as soon as possible or in a specific week.
- **Frozen period.** When planning based on production status or production date, Isah can take into account a frozen period (days) to ensure that POs within the frozen period are not changed.

- **Planning horizon.** When planning based on production status/-production date, Isah can use a planning horizon to schedule all POs within the chosen planning horizon.
- **Priorities.** Isah can use one or more priorities while planning. When using more priorities, the priorities are ranked. For each priority, it should be defined if it should be used in ascending or descending order. The following priorities can be used in Isah: customer priority, sales priority, shipment date of the sales order, sales status, PO end date, production priority, and production status.

  Figure 2.8 shows an example of using priorities in Isah. In this example, first, all POs in a specific production status (in Dutch: *Productiestatus*) with a customer priority (in Dutch: *Klantprioriteit)* of 1 are planned based on ascending (in Dutch: *Oplopend*) sales order shipment date (in Dutch: *Verzenddatum*). When production statuses have the same customer priority and sales order shipment date, the highest production status will be planned first since it uses a descending (in Dutch: *Aflopend*) order. All POs without a customer priority are planned last based on the sales order shipment date.

This Figure is not publicly available due to confidentiality.

Figure 2.8: Example of priorities used within Isah.

- **Capacity overview.** It is possible to analyse the capacity of the operations or employees to identify capacity bottlenecks in the created production plan. The capacity overview is shown in a Gantt Chart.

2.4 Performance of current due date setting and operational planning method of HA

The following sections describe the performance of the current planning method in terms of Customer Service Level (2.4.1) and tardiness and earliness (2.4.2).

2.4.1 Customer Service Level

Customer Service Level is the KPI that initiated this research. See Section 1.2 Table 1.1 for the Customer Service Level of HA from 2017-2022. Section 1.2 also explains the calculation of the Customer Service Level and the aim of management regarding the Customer Service Level.

2.4.2 Tardiness and earliness

KPIs used in production are tardiness and earliness. A tardy PO is a PO where the completion date is higher than the internal due date, $T_i = max(0, C_i - d_i)$ (Muñoz-Villamizar et al., 2019). An early PO is a PO where the completion date is lower than the internal due date, $E_i = max(0, d_i - C_i)$ (Muñoz-Villamizar et al., 2019). The deviation between the completion date and the internal due date is measured in days. Table 2.7 shows the percentages of early POs, POs on-time and tardy POs at HA from 2017-2022.

Table 2.7: Percentages of early POs, POs on-time and tardy POs of HA.

	2017	2018	2019	2020	2021	2022
% of early POs	36.3%	36.4%	1.5%	35.7%	47.7%	30.4%
% of POs on time	8.7%	31.3%	28.9%	16.2%	25.6%	20.3%
% of tardy POs	55.0%	32.2%	39.6%	48.1%	26.7%	49.3%

We cannot see a constant improvement or setback for the percentages over the years, which is logical since planning is conducted in the same manner in all these years. The percentages are not in line with the goal of management. Theoretically, HA wants 100% POs on time. The more realistic aim of management is 90% POs on time, 8% early POs and 2% tardy POs. Early POs are better than tardy POs and therefore HAs aim at a percentage of 8% of early POs and 2% of tardy POs. However, there are several reasons why HA wants its percentage of early POs as low as possible:

- Not all customers want to receive the products before the agreed delivery date. Therefore, products of early POs are kept in inventory until the delivery date reaches. Keeping finished products in inventory costs money and increases the inventory position.
- Early POs can indicate that internal lead times for the product are lower than expected. This means that the start of PO could be later and this results in the later purchasing of components. This would lead to a lower inventory position since components would be in stock for a shorter period.

Improving the inventory position of HA is outside the scope of this research. The remainder of this research only uses tardiness as KPI.

2.5 Conclusion on the current situation

HA operates in an MTO environment. It has an SMT and THT production method that can perform 3 and 14 unique operations, respectively. The SMT production is a flow shop since each product follows the same production route. The THT department is a job shop where each product has its own predetermined production route. The entire production environment of HA is analysed as a job shop to generalise this research.

The due date setting of a sales order is executed by the operations manager based on the customer's preferred due date, the existing production plan, component availability and sales order priority. The amount of resource capacity available is based on the experience and gut feeling of the operations manager instead of the resource capacity available in the currently available production plan. This leads to unrealistic external due dates.

After a confirmed sales order the operations manager creates POs for the SMT and THT department. These POs are planned with an automatic work preparation function in Isah. Afterwards, the POs are dispatched to the shop floor. The SMT department dispatches based on the earliest starting date of a PO. The THT department dispatches based on the earliest external due date.

This research improves the process of planning the created POs. Instead of creating an internal due date for THT equalling the first confirmed due date of the sales order minus 2 slack days and for SMT equalling the start date of THT minus 7 slack days, we design a method that sets internal due dates for the PO based on the external due date of the sales order and resource capacity available. The designed method takes into account the production environment characteristics found in this chapter.

3 Literature review

Section 1.4 defines the scope of this research: resource loading. So, the literature review focuses on resource loading. Resource loading measures the impact of a set of orders for a specific set of time buckets (Hans, 2001). Section 3.1 describes the concept of resource loading. Section 3.2 describes solution approaches for optimisation problems such as resource loading. Section 3.3 provides a conclusion to the literature study.

3.1 Resource loading

Resource loading is a tactical planning method often used at order acceptance. It measures the impact of a set of orders on the production system (Hans, 2001). It loads a given set of orders and determines reliable internal due dates and resource capacity levels needed to process the orders and their constituting jobs (Hans, 2001). Resource loading can be executed simultaneously (integrated approach) or separately (hierarchical approach) from the order acceptance decision. In the hierarchical approach, the only information passed through from the order acceptance decision to resource loading is the aggregate information on the workload (Hans, 2001).

Hans (2001) mentions two resource loading approaches: a time-driven approach and a resource-driven approach. The resource-driven approach allows orders to be finished after their due date because capacity cannot be extended. Common objectives mentioned are minimising total lateness or minimising the number of orders too late. The time-driven approach extends short-term capacity against a certain cost to ensure that orders are finished before their due dates. The objective function is minimising the cost of extending capacity. In general, three ways of capacity expansion are used: overtime, hiring extra operators and outsourcing.

Hans (2001) provides a Mixed-Integer Linear Programming model (MILP) for the resource loading problem. This model adopts the time-driven and resource-driven approaches simultaneously. The model takes into account fixed machine capacity and expandable operator capacity. The orders of an MTO job shop production environment are divided into jobs with linear precedence relations (i.e. a job may start only if its predecessor is completed). Figure 3.1 shows an example of a linear precedence relation. The model can handle a minimal duration of a job and a minimum time lag between jobs. The model uses time buckets of one week. The release and due dates of jobs orders are specified in weeks and the processing times of jobs are specified in hours. The model minimises the total cost of extending capacity and the penalties incurred by tardy orders.

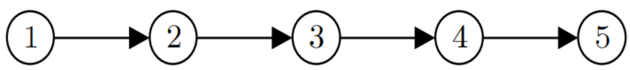

Figure 3.1: Example of linear precedence relations as provided by Hans (2001).

An equivalent approach to resource loading is also used in project management: Multi-Project Rough-Cut Capacity Planning (RCCP). RCCP allows generalised precedence constraints, which implies that they do not have to be linear. Figure 3.2 shows an example of generalised precedence relations. Gademann and Schutten (2005) study a time-driven RCCP taking into account capacity flexibility, precedence relations and maximum work content per period. They discuss several linear programming-based heuristics to solve their model. Cherkaoui et al. (2015) propose a time-driven RCCP MILP model that handles different planning levels by varying the length of the time periods. Their model is based on a continuous time representation of start and end times and the discrete-time representation of resource constraints. The period lengths are shorter at the beginning of the planning horizon and increase over time.

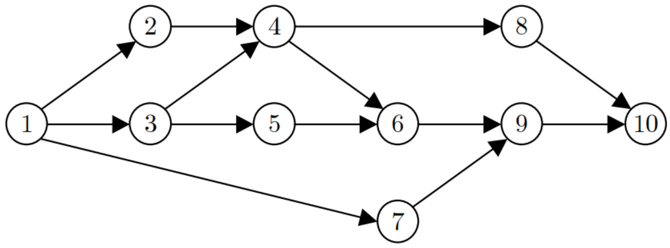

Figure 3.2: Example of generalised precedence relations as provided by Hans (2001).

3.2 Solution approaches for resource loading

Two classes of optimisation algorithms exist: exacts methods and heuristic methods (Rader, 2010). Exact methods provide optimal solutions. Examples of exact methods are complete enumeration, branch-and-bound and simplex (Guzman et al., 2022; Schutten, 1996). For most applications, exact methods are too slow to solve the problem in a reasonable period (Guzman et al., 2022). The remainder of this study focuses on heuristic methods since Section 3.1 already provides an exact method: a MILP model. Heuristic methods are mainly divided into two classes: constructive heuristics and improvement heuristics (Guzman et al., 2022; Maan-Leeftink, 2021a).

3.2.1 Constructive heuristic

Constructive heuristics generate solutions incrementally by building upon a partial (incomplete) solution (Maan-Leeftink, 2021a; Tavares et al., 2009). Constructive heuristics require little computational time and are easy to implement (Schutten, 1996). One major disadvantage is that they lead to poor-quality solutions. However, they can be used to create an initial solution to use for improvement heuristics. One way to add building blocks to an empty solution is based on priority rules such as Earliest Due Date, Shortest Processing Time and First In-First Out (Schutten, 1996). In due-date setting literature, another priority rule is mentioned that not only determines a feasible solution based on job information but also based on capacity constraints (Thürer et al., 2013). This rule is known as finite loading.

Priority rules. Priority rules examine the order in which orders are scheduled by assigning priorities. Some well-known priority rules in literature are Earliest Due Date (EDD), First In First Out (FIFO), Minimal Slack (MS), Service in Random Order (SIRO) and Shortest Processing Time (SPT) (Holthaus and Rajendran, 1997; Pinedo, 2016a; Schutten, 1996).

Finite loading. Finite loading can be executed forward or backwards. Forward Finite Loading (hereinafter referred to as FFL) is recommended when orders have negotiable due dates because they are proposed or quoted by the company (Thürer et al., 2013). Backward Finite Loading (hereinafter referred to as BFL) is recommended when orders have fixed due dates specified by the customer (Thürer et al., 2013). Assume order i with product routing R_i consisting of n_i jobs , i.e. $R_i = \{1, ..., n_i\}$. All jobs of order i need to be planned on machine 1 ($s = 1$) in time buckets $t = \{1, 2, 3, 4, 5, 6\}$. FFL starts at the release date of the first job r_{i1} and plans each job j accordingly. Due date d_{ij} is set to the due date of the previous job $d_{i(j-1)}$ plus its processing time p_{ij} and the dynamic factor $F_{ij}(W_{st}, C_{st})$. $F_{ij}(W_{st}, C_{st})$ is based on the workload W_{st} and capacity C_{st} of machine s in time bucket t. For example, when $d_{(i-1)j} = 4$ and $p_{ij} = 1$ we load (i, j) in time bucket $4 + 1 = 5$ since W_{15} + workload $(i, j) \leq C_{15}$. Equation 3.1 shows the formula for FFL for each i except for the first one. For the first operation in R_j, $d_{i-1,j}$ is changed by the release date of the order r_i. Figure 3.3 shows a graphical representation of the example provided.

$$d_{ij} = d_{i,j-1} + p_{ij} + F_{ij}(W_{st}, C_{st}) \qquad \forall i\ \forall j \in R_i \setminus \{1\} \tag{3.1}$$

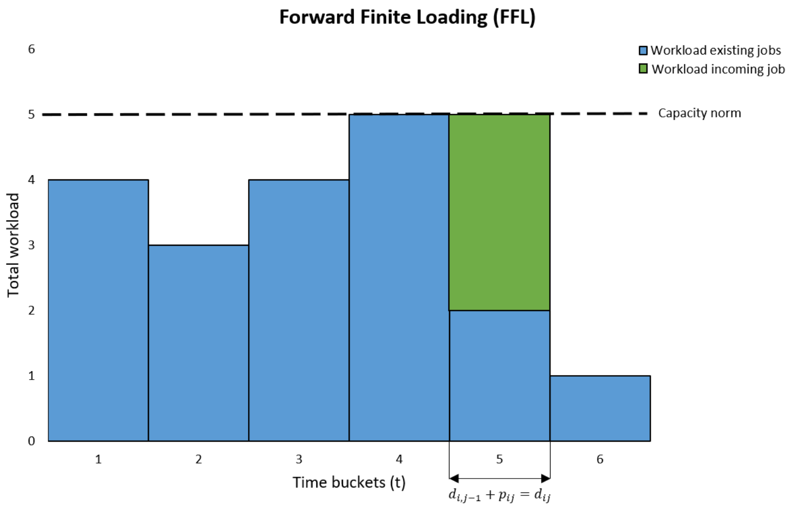

Figure 3.3: Graphical representation of FFL adjusted from Thurer et al. (2014).

With BFL we start at the due date of order i. Equation 3.2 shows BFL for all operations in R_i except for the last operation n_i. The due date of the last operation n_i in product routing R_i is set to the due date of the order.

$$d_{ij} = d_{i(j+1)} - p_{ij} - F_{ij}(W_{st}, C_{st}) \qquad \forall i\ \forall j \in R_i \setminus \{n_i\} \tag{3.2}$$

For FFL and BFL the size of the time buckets must be larger than the largest processing time p_{ij} so that an operation can always be planned in a single time bucket (Robinson and Moses, 2006). If the processing time variability is high then the required minimum bucket size is large relative to the average processing time. Robinson and Moses (2006) proposes a method that overcomes this problem: Partially Forward/Backward Finite Loading (PFFL/PBFL).

PFFL/PBFL first calculates the number of time buckets required to accommodate an operation. Equation 3.3 shows the calculation for the number of time buckets where b is the number of time buckets required, p_{ij} is the processing time and G is the granularity of the time buckets.

$$b = \frac{p_{ij}}{G} \tag{3.3}$$

If $b = 1$, FFL or BFL is used. If $b > 1$, PFFL or PBFL looks for availability across multiple contiguous time buckets. This means that after finding a time bucket the procedure checks the subsequent $b - 1$ time buckets. To avoid waiting time, the procedure will not insert a job unless each of the subsequent time buckets is 100% available. If the availability of contiguous time buckets is insufficient, the search will start again at the next time bucket. Figure 3.4 shows a graphical representation of PFFL where (i, j) is partially loaded in time buckets 1 and 2.

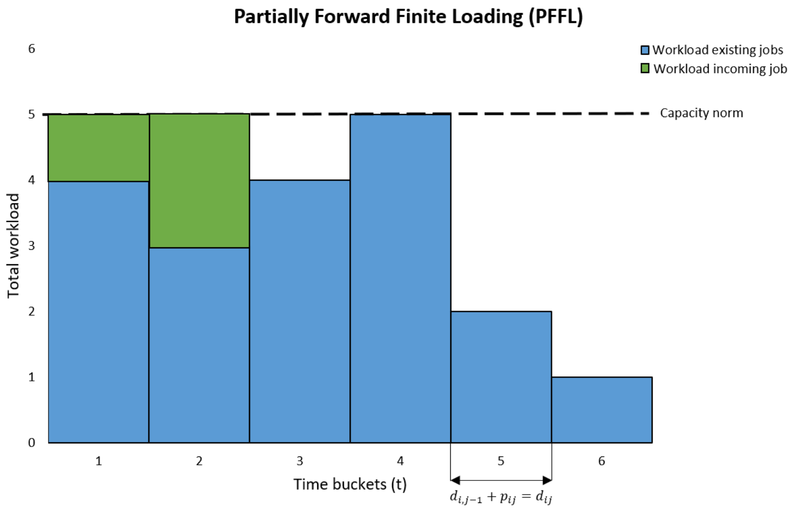

Figure 3.4: Graphical representation of PFFL adjusted from Robinson and Moses (2006).

3.2.2 Improvement heuristics

Improvement heuristics apply improvements in each iteration to a complete initial solution until a stopping criterion is met (Tavares et al., 2009). The initial solution is generated randomly or by a constructive heuristic (Tavares et al., 2009). One type of improvement heuristic is local search. Local search starts with a given initial feasible solution, searches the neighborhood for a better solution using some operator, accepts/rejects the solution and starts again. The local search stops if a local optimum or stopping criteria is reached (Maan-Leeftink, 2021b). A local optimum is the best solution found in the neighborhood but it does not have to be the best solution of the entire feasible region (i.e. global optimum). Figure 3.5 shows this concept.

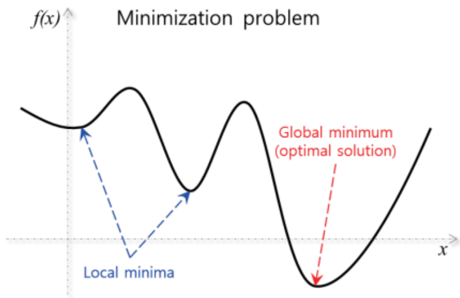

Figure 3.5: Illustration of local and global optimality by Jeon et al. (2017).

To overcome local optimality a balance between intensification and diversification is used. Intensification exploits a neighborhood with promising solution characteristics and diversification explores the entire feasible region (Maan-Leeftink, 2021c). Heuristics that balance intensification and diversification are called metaheuristics (Maan-Leeftink, 2021c). Two well-known metaheuristics are Simulated annealing (hereinafter referred to as SA) and Tabu Search (hereinafter referred to as TS).

Simulated Annealing. SA is able to escape from local optima by accepting changes to the solution (with a certain probability) that worsen the objective function. SA starts with an initial solution based on a (greedy) constructive heuristic. At each iteration, a neighbor solution is generated randomly by a neighborhood operator. Some examples of neighborhood operators are move, swap and reassign (Maan-Leeftink, 2021c). The neighbor solution is accepted as the new current solution if the objective function is better. If the objective function is worse, the neighbor solution is accepted with a certain probability. A temperature is used to influence the probability of accepting a worse solution. SA starts with an initial temperature. After each iteration, the temperature decreases via a cooling scheme. A lower temperature causes a lower acceptance rate of a worse solution. SA stops if a stopping condition is met. Rules of thumb are defined in literature to set the parameters for SA (Maan-Leeftink, 2021c; Rader, 2010):

- **Initial temperature:** Choose an initial temperature such that the initial acceptance ratio of worse solutions is approximately 1. This leads to high diversification at the start of SA. The acceptance ratio is calculated by $\frac{\text{number of accepted worse transitions}}{\text{number of proposed worse transitions}}$.
- **Length of Markov chains:** After each Markov Chain the temperature is decreased such that the probability of accepting a worse solution becomes smaller until the probability becomes almost zero. The number of iterations is determined by the Markov Chain length. E.g. a Markov Chain length of 1 means that after every iteration the temperature is updated. The Markov Chain length can be static or dynamic. A rule of thumb is to set the Markov Chain length equal to the number of neighbor solutions.
- **Cooling schemes:** The cooling scheme defines how fast the temperature decreases. Two commonly used examples are $NewTemperature = \alpha * CurrentTemperature$ and $NewTemperature = \frac{CurrentTemperature}{1+\beta * CurrentTemperature}$ with α close to 1 and β close to 0.
- **Stopping condition:** A few examples of stopping criteria are (1) reaching a maximum number of iterations, (2) temperature T getting close to 0, and (3) the current solution does not change after too many iterations. The most used stopping criterion is the temperature getting close to 0.

Tabu search. TS is able to escape from local optima by incorporating memory. TS also starts with an initial solution based on a (greedy) constructive heuristic. Based on the initial solution, (a selection of) neighbor solutions are evaluated and accepted if their objective value is better and is not on the tabu list (Maan-Leeftink, 2021b). Attributes from recently visited solutions are put in the front of the tabu list and the last element is deleted. Some important decisions when implementing TS (Maan-Leeftink, 2021b; Rader, 2010):

- **Tabu list length:** The tabu list length can be static or dynamic. With a dynamic length, the length increases if the best solution is not updated and decreases if a certain time has passed without solution revisits.
- **Tabu list attributes:** Possible attributes are (1) entire solutions, (2) operators, and (3) changed items. The best attribute to use depends on the problem and its size.
- **Aspiration level:** A disadvantage of some tabu list attributes is that multiple solutions can be rejected due to one stored attribute, also unvisited once. Aspiration level is a rule allowing changes that are in the tabu list. One example is accepting the neighbor solution if it leads to the best solution found.
- **Stopping criteria:** The most common stopping criteria is no improvements after a predetermined number of exchanges or time.

3.3 Conclusion on the literature study

Resource loading is used to measure the impact of a set of orders in terms of due dates and resource capacity levels. Resource loading can be time-driven, resource-driven or a combination of both. Hans (2001) provides a MILP for the combined resource loading problem. An equivalent approach allowing generalised non-linear precedence constraints is Rought-Cute Capacity Planning (RCCP). Gademann and Schutten (2005) provide a time-driven RCCP.

Solution approaches for resource loading are exact methods and heuristic methods. Heuristic methods are mainly divided into two classes: constructive heuristics and improvement heuristics. Constructive heuristics creates a solution incrementally by building upon a partial (incomplete) solution. Constructive heuristics are easy to implement and need less computational time but the solution quality is low. However, the solution of constructive heuristics can be used as input for the improvement heuristic. Examples of constructive heuristics are simple priority rules (e.g. Earliest Due Date, Shortest Processing Time) and priority rules taking into account capacity (i.e. finite loading). Improvement heuristics apply improvements in each iteration to a complete initial solution until a stopping criterion is met. Examples of improvement heuristics are local search, Simulated Annealing and Tabu Search.

4 Solution design

This chapter proposes a resource loading MIP formulation and a resource loading heuristic consisting of a construction and improvement step. Section 4.1 describes the model formulation. Section 4.2 describes the designed constructive heuristic. Section 4.3 describes the designed improvement heuristic. Section 4.4 provides a conclusion to this chapter.

4.1 MIP model formulation

This section describes the resource-loading MIP model formulation. Section 4.1.1 describes the sets, parameters and decision variables. Section 4.1.2 describes the objective value and constraints.

4.1.1 Sets, parameters and decision variables

This section provides a description and the assumptions of the job shop production environment of HA. The time horizon T is discretized into time buckets ($t = T_{min}, T_{min+1}, T_{min+2}, ..., T_{max-1}, T_{max}$) of equal length. The production environment of HA knows 17 different operations J_j ($j = 1, ..., 20$). Table 4.1 shows the different operations J_j. Each operation has regular capacity c_j (hours) per time bucket. HA is able to extend capacity only by working overtime. Per time bucket a maximum amount of overtime mo_{jt} is allowed for each operation. Decision variable O_{jt} indicates the overtime (hours) for operation J_j in time bucket t. Working overtime has a cost of $\overline{co}$ per hour. The total cost of working overtime is given by $\sum_{jt} \overline{co} * O_{jt}$.

Table 4.1: Overview of the operations at HA in the set J_j.

j	**Operation**	j	**Operation**
1	AOI	*10*	PWA
2	COAT	*11*	REW
3	EC	*12*	SMD
4	PB	*13*	SOL
5	POT	*14*	SSOL
6	PR	*15*	TC
7	PRNA	*16*	TE
8	PRO	*17*	VER
9	PW		

Each order I_i ($i = 1, ..., n$) has a product routing consisting of z_i operations. The precedence constraints are linear, this means that the product routing shows the operations to be performed in a given sequence. For example, the product routing of order I_1 is $J_1 \rightarrow J_3 \rightarrow J_4$, whereas the product routing of order I_2 is $J_4 \rightarrow J_5 \rightarrow J_2 \rightarrow J_6$. S_{is} indicates the s-th operation in the product routing of order I_i. For example, S_{11} indicates the first operation in the product routing of order I_1 which equals J_1. S_{23} indicates the third operation in the product routing of order I_2 which equals J_2. Operation S_{is} has a pre-defined positive processing time p_{is} (hours). There are two options: (1) we can process an operation in one time bucket or (2) we need multiple time buckets to process the entire operation. For option 1, we need a time bucket size large enough such that the largest operation can always be planned in one time bucket. Option 2 is recommended when having a high processing time variability. We use option 2 in this research. In 2022 the longest operation at HA needed six weeks to finish. Considering a time bucket size of six weeks is impractical for HA and also not preferred by the management. Most operations at HA take a few hours or days, and a small percentage of the operations take more than one week (1.5% in 2023). We use a time bucket of one week. To be able to cope with the more time-consuming orders, we allow multiple time buckets to process the entire operation. x_{ist} indicates the number of hours that the s-th operation of order I_i is assigned to time bucket t. Binary variable y_{ist} indicates if the s-th operation of order I_i is assigned to time bucket t. HA does not allow preemption within an operation of order I_i, i.e. the execution of operation $s \in S_i$ is planned in subsequent weeks.

The orders at HA have an internal due date d_i (expressed in weeks) but no release date. To simplify the model, we assume that all materials are available before the start of the time horizon. Orders are allowed to finish earlier or later than the internal due date. $Tardy_i$ indicates if order I_i is finished too late. Finishing later than the internal due date is allowed against some cost $\overline{ct}$. The cost of finishing late is independent of the number of days finishing late. The total cost of tardy orders is given by $\sum_i \overline{ct} * Tardy_i$. Table 4.2 shows an overview of all sets, parameters and decision variables.

A solution is feasible if each operation in S_i is completely assigned within the planning horizon while respecting the precedence, capacity and non-preemption constraints.

Table 4.2: Sets, parameters and decision variables

Sets	
I_i	Set of all orders, indexed by i
J_j	Set of all operations, indexed by j
T	Set of all time buckets, indexed by t where $t = \{T_{min}, T_{min+1}, T_{min+2}, ..., T_{max-1}, T_{max}\}$
S$_i$	Product routing of order i, indexed by s where $s = \{1, 2, ..., z_i\}$
Parameters	
S_{is}	The s-th operation in the product routing of order i
p_{is}	Processing time of the s-th operation of order i (expressed in hours)
d_i	Internal due date of order i (expressed in weeks)
c_j	Regular capacity of operation j (expressed in hours)
mo_{jt}	Maximum overtime for operation j in time bucket t (expressed in hours)
$\overline{ct}, \overline{co}$	Cost of tardiness and overtime, respectively
Decision variables	
x_{ist}	Number of hours that the s-th operation of order i is assigned to time bucket t
y_{ist}	1 if the s-th operation of order i is processed in time bucket t, 0 otherwise
O_{jt}	The hours of overtime used for operation j in time bucket t
$Tardy_i$	1 if order i is finished too late, 0 otherwise
Auxiliary variables	
s_{ist}	Auxiliary variable used for non-preemption constraints 4.7 to 4.11 1 if the job started in or before time bucket t, 0 otherwise
f_{ist}	Auxiliary variable used for non-preemption constraints 4.7 to 4.11 1 if the job is not completed before time bucket t, 0 otherwise

4.1.2 Objective function and constraints

The goal of HA is to minimise the cost of overtime and tardiness. The objective function equals:

$$Min(z) = \sum_{i} \Big(\overline{ct} * Tardy_i \Big) + \sum_{jt} \Big(\overline{co} * O_{jt} \Big)$$

Constraint (4.4) ensures that each operation in S_i is fully assigned. The operation may be assigned preemptive or non-preemptive. Constraints (4.7) until (4.11) together with Constraint (4.4) ensures non-preemption within an operation of order i.

$$\sum_{t} x_{ist} = p_{is} \qquad \forall i, s \in S_i \tag{4.4}$$

The binary decision variable y_{ist} equals 1 if the s-th operation of order i is processed in time bucket t, i.e. if $x_{ist} > 0$. Constraints (4.5) and (4.6) ensure that $y_{ist} = 1$ if $x_{ist} > 0$. If $x_{ist} > 0$, constraint (4.5) holds for $y_{ist} = 0 \vee 1$ while constraint (4.6) forces y_{ist} to be 1. If $x_{ist} = 0$, constraint (4.6) holds for $y_{ist} = 0 \vee 1$ while constraint (4.5) forces y_{ist} to be 0. Epsilon is set to the minimum duration required to start S_{is}. For example, if x_{ist} needs to be at least 30 minutes ϵ is set to 0.5.

$$x_{ist} \geq y_{ist} * \epsilon \qquad \forall i, s \in S_i, t \tag{4.5}$$

$$x_{ist} \leq p_{is} * y_{ist} \qquad \forall i, s \in S_i, t \tag{4.6}$$

Constraints (4.7) until (4.11) (in combination with constraint (4.5)) ensures non-preemption within an operation of order i. Binary variables s_{ist} and f_{ist} indicate the start- and end time of the s-th operation of order i. s_{ist} equals 1 if the job started in or before time bucket t and 0 if the job has not started yet. f_{ist} equals 1 if the job is not completed before time bucket t and 0 if the job is completed before time bucket t. Table 4.3 provides an example of the logic between y_{ist}, s_{ist} and f_{ist}. s_{ist} equals 1 for all t starting from the first t in which $y_{ist} > 0$. f_{ist} equals 1 starting from $t = 1$ and becomes zero as soon as y_{ist} becomes zero again. Constraints (4.7) and (4.8) set s_{ist}, whereas constraints (4.9) and (4.10) set f_{ist}. Constraint (4.11) ensures that y_{ist} equals one if both s_{ist} and f_{ist} equal 1.

Table 4.3: Example of auxiliary variables non-preemption constraints.

t	1	2	3	4	5	6
y	0	0	1	1	1	0
s	0	0	1	1	1	1
f	1	1	1	1	1	0

$$s_{ist} \geq y_{ist} \qquad \forall i, s \in S_i, t \tag{4.7}$$

$$s_{ist} \geq s_{is(t-1)} \qquad \forall i, s \in S_i, t \tag{4.8}$$

$$f_{ist} \geq y_{ist} \qquad \forall i, s \in S_i, t \tag{4.9}$$

$$f_{ist} \geq f_{is(t+1)} \qquad \forall i, s \in S_i, t \tag{4.10}$$

$$y_{ist} \geq s_{ist} + f_{ist} - 1 \qquad \forall i, s \in S_i, t \tag{4.11}$$

Constraint (4.12) ensures that the assigned hours for operation j in time bucket t may not exceed the regular capacity c_j plus the available overtime O_{jt}. For example, order 1 has product routing $J_2 \rightarrow J_4$ and order 2 has product routing $J_8 \rightarrow J_4 \rightarrow J_1$. Operation J_4 is performed at S_{12} and S_{22}. The resulting capacity check for operation J_4 equals: $x_{12t} + x_{22t} \leq c_4 + o_{4t}$ for all time buckets t. Constraint (4.13) ensures that maximum overtime is not exceeded.

$$\sum_{i} \sum_{\{s \in S_{is} | S_{is} = j\}} x_{ist} \leq c_j + O_{jt} \qquad \forall j, t \tag{4.12}$$

$$O_{jt} \leq mo_{jt} \qquad \forall j, t \tag{4.13}$$

The linear precedence constraint assures that the s-th operation in the product routing can start if $(s-1)$ is finished. This should hold for every s in S_i except for $s = 1$. Equation (4.14) shows the corresponding constraint.

$$\sum_{t'=1}^{t} x_{i(s-1)t'} \geq p_{i(s-1)} * y_{ist} \qquad \forall i, t \; \forall s \in S_i \setminus \{1\} \tag{4.14}$$

Constraint (4.15) indicates if an order is tardy. If the last operation z_i in product routing S_i of order i is finished in time bucket t higher than d_i, the order is tardy (i.e. $Tardy_i = 1$).

$$Tardy_i \geq y_{ist} \qquad \forall i \forall \{s \in S_i | s = z_i\} \forall \{t \in T | t > d_i\} \tag{4.15}$$

Constraint (4.16) shows the non-negative sign variables and constraint (4.17) shows the binary sign variables.

$$x_{ist}, Overtime_{jt} \geq 0 \tag{4.16}$$

$$y_{ist}, s_{ist}, f_{ist}, Tardy_i \in \{0, 1\} \tag{4.17}$$

4.2 Constructive heuristic

This section describes the designed constructive heuristic. The constructive heuristic is used to construct an initial feasible solution. Section 3.2.1 describes four finite loading methods: FFL, BFL, PFFL and PBFL. We choose the idea of PBFL for two reasons:

1. External due dates of orders are established at order acceptance and are fixed.
2. Operations are allowed to be assigned to several consecutive time buckets.

PBFL looks backwards for availability across multiple contiguous time buckets. A job is not inserted unless each of the subsequent time buckets is 100% available. The constructive heuristic starts with an empty solution and plans the orders according to a sequence. First, the sequence is determined based on a priority rule (see Section 4.2.1). Afterwards, the order is planned using PBFL (see Section 4.2.2). Each order with its operations is planned backwards while taking into account precedence, capacity and non-preemption constraints. The constructive heuristic is finished (1) when a feasible solution is found or (2) when the problem is infeasible. The problem is infeasible when the orders cannot be planned within the planning horizon while respecting the precedence, capacity and non-preemption constraints. Figure 4.1 shows a graphical representation of the constructive heuristic.

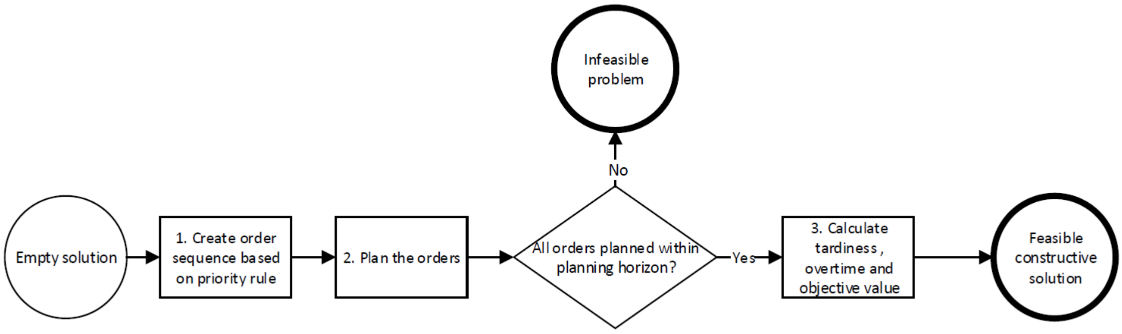

Figure 4.1: Flowchart of the constructive heuristic.

Section 4.2.1 describes the priority rules considered in the constructive heuristic. Section 4.2.2 describes how the orders are planned and Section 4.2.3 describes the calculation of the objective value.

4.2.1 Select order based on priority rule

The sequence in which orders are planned is determined by a priority rule. Section 3.2.1 discusses well-known priority rules. The priority rules implemented in the constructive heuristic are Earliest Due Date (EDD), Latest Due Date (LDD), Shortest Processing Time (SPT) and Longest Processing Time (LST). With EDD and LDD, we determine if it is better first to schedule all early or late internal due dates when planning backwards. With SPT and LST we determine if it is better to first schedule smaller or larger orders.

4.2.2 Plan orders

The orders are planned in (consecutive) time bucket(s) with capacity or overtime available to plan operation S_{is}. This is executed until the complete processing times of all orders are planned. Figure 4.2 shows a graphical representation of step 2 in Figure 4.1. To be able to search in multiple time buckets for availability, we introduce an Available-To-Work (ATW) window as used in Gademann and Schutten (2005). The following paragraphs provide a description of setting the ATW windows (step 2.1. in Figure 4.2) and ensuring planning in (consecutive) time bucket(s) (step 2.2. in Figure 4.2).

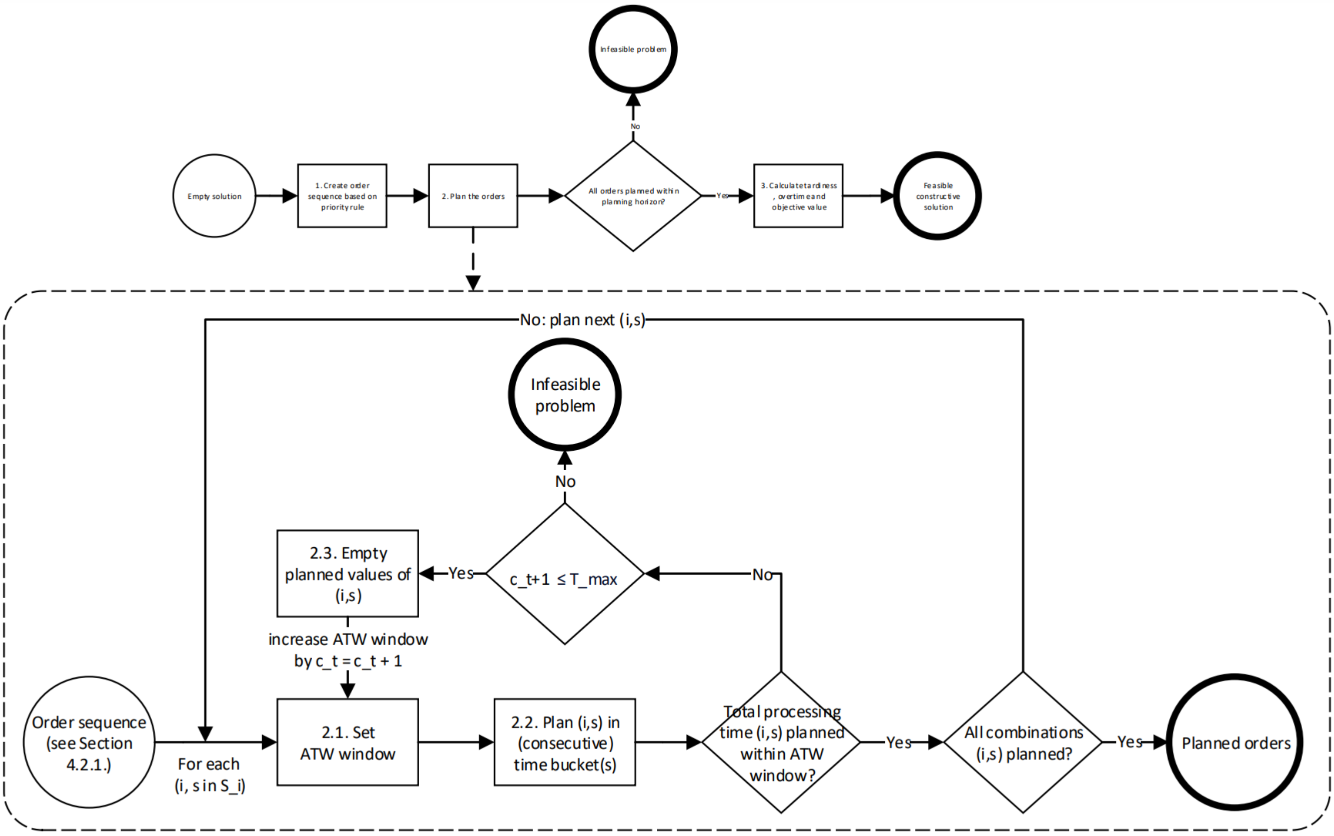

Figure 4.2: Flowchart of step 2 (plan orders) in the constructive heuristic.

Step 2.1: Set ATW window. The ATW window $[F_{is}, L_{is}]$ indicates the range in which we look for available (consecutive) time bucket(s) to plan S_{is}. F_{is} indicates the first time bucket and L_{is} indicates the last time bucket in which we look for available capacity. The ATW window loops negatively in time. The ATW window for the last operation z_i starts at the internal due date d_i until the minimum time bucket in the planning horizon, i.e. $[d_i, T_{min}]$. For example, the internal due date is in time bucket 8 and the minimum time bucket of the planning horizon is 6. The ATW window is set to [8,6], i.e. we loop through the time buckets in the following sequence $8 \rightarrow 7 \rightarrow 6$. The ATW window for all other operations in S_i starts at the start time of the next operation until the minimum time bucket in the planning horizon, i.e. $[st_{i(s+1)}, T_{min}]$.

If the total processing time p_{is} does not fit in the ATW window, the ATW window is enlarged by increasing L_{is} by one. To avoid violation of the precedence constraint, all set planning variables of S_i are emptied and step 2.2. is executed again for all $s \in S_i$. Increasing the ATW window is executed until T_{max} is reached. If T_{max} is reached and the complete p_{is} cannot be planned within the ATW window, the problem is infeasible.

Step 2.2: Plan S_{is} in (consecutive) time bucket(s). Within the ATW window, one or multiple consecutive time buckets are selected to plan S_{is}. For each time bucket in the ATW window, we check if there is either regular capacity or overtime available. The constructive heuristic is allowed to plan multiple operations of one order in the same time bucket. This leads to the following situation:

if operation S_{is} is scheduled for 60% of the time bucket, operation S_{is+1} cannot be scheduled for more than 40% of the time bucket. To prevent this situation, the auxiliary parameter $PercentageWeek_{it}$ is initialised to ensure that the percentage of planned order I_i in time bucket t does not exceed 100%. If there is (1) not enough regular capacity or overtime available to plan the complete processing time in time bucket t or (2) $PercentageWeek_{it} \geq 1$, we check if the consecutive time bucket $t-1$ has regular capacity or overtime left. If part of an order is planned in time bucket t and time bucket $t-1$ has no regular capacity or overtime left or $PercentageWeek_{it} \geq 1$, we delete all planned hours of S_{is} and start planning again at the next time bucket $t-1$ such that consecutiveness is ensured. This is executed until (1) the complete processing time p_{is} is scheduled within the ATW window or (2) every time bucket in the ATW window is evaluated. Figure 4.3 shows a graphical representation of step 2.2. (plan (i,s) in (consecutive) time bucket(s)). Appendix A provides a technical description of the flowchart in Figure 4.3.

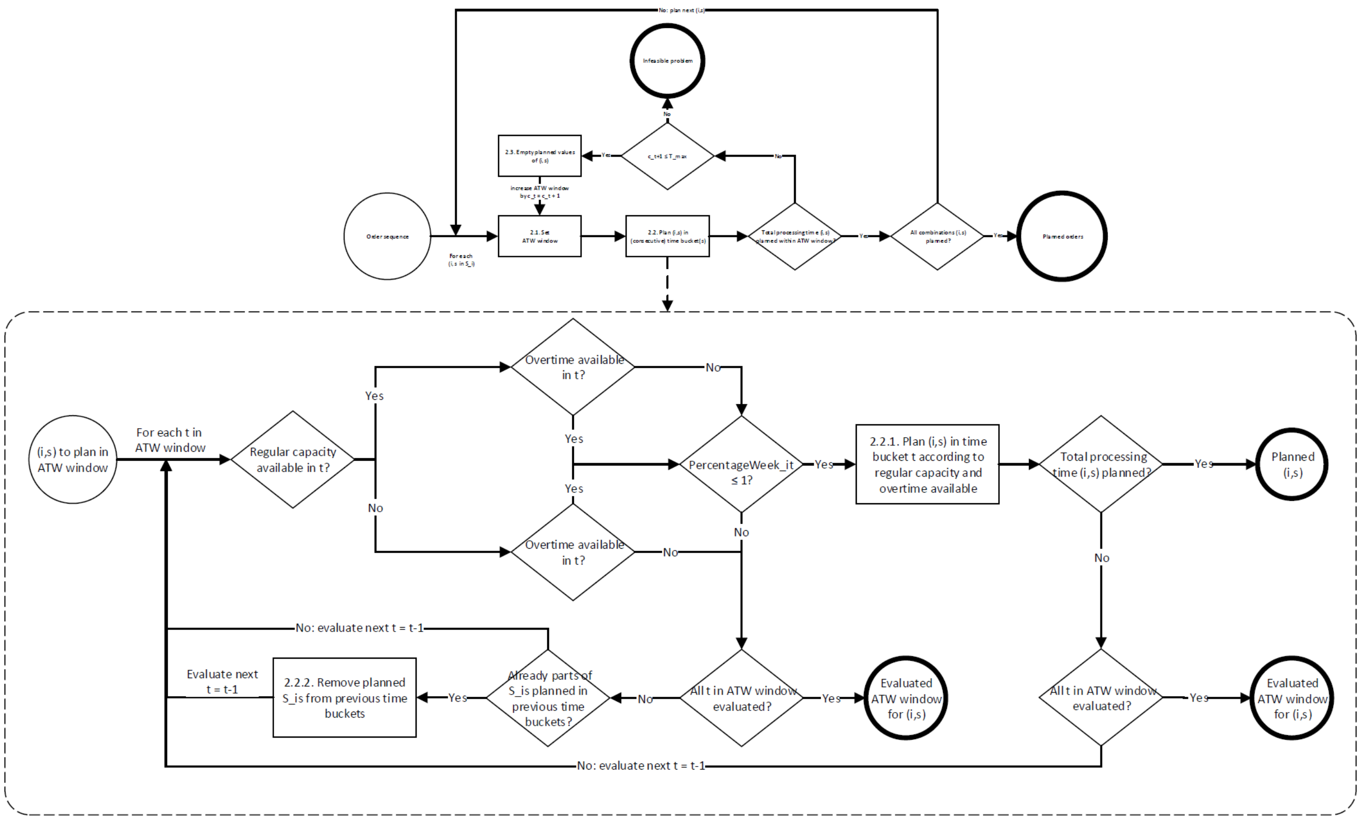

Figure 4.3: Flowchart of step 2.2. (plan (i,s) in (consecutive) time bucket(s)) of the constructive heuristic.

4.2.3 Calculate objective value

Tardiness and overtime are calculated when all orders with their operations are completely planned. One limitation of the constructive heuristic is that it only calculates tardiness and does not make decisions on tardiness. For example, if an operation in $t = d_i$ needs 10 hours of overtime with a total cost of €200 while planning in $t+1$ does not require overtime but leads to a tardy order with a total cost of €100, the constructive heuristic plans the operation in $t = d_i$. The improvement heuristic improves the constructive heuristic by checking if it is better to decrease overtime by increasing tardiness and the other way around. One way to improve the constructive heuristic is to make decisions on tardiness versus overtime. So, in the previous example, the constructive heuristic then plans at $t+1$ instead of t. This is a recommendation for future research (see Section 6.2).

4.3 Improvement heuristic: Simulated Annealing

After constructing an initial feasible solution, SA is used to improve the initial solution. We choose SA over TS since it explores and evaluates one neighbor at a time instead of multiple neighbor solutions (Aarts et al., 1992). At TS, each neighbor solution needs to be evaluated against the solutions already in the tabu list. SA also needs less tailored parameter tuning (Aarts et al., 1992). At the beginning of the improvement heuristic, a set is created from the initial solution containing tardy orders and orders using overtime. At each iteration of the improvement heuristic, a random order i is selected from the created set. A neighbor is created with the neighborhood operator using the randomly selected order. If the objective value of the neighbor is better than the current solution, the neighbor is accepted as the new current solution. If the objective value of the neighbor is also better than the current best solution, the neighbor is accepted as the new current best solution. If the objective value of the neighbor is worse than the current solution, the neighbor is accepted with a certain probability. After each Markov Chain, the temperature is decreased. Algorithm 1 provides the pseudo-code of the heuristic. Section 4.3.1 describes the neighborhood operator of the SA heuristic.

Algorithm 1 Improvement heuristic: Simulated Annealing

```
Temp := InitialTemp
Solution := ConstructiveSolution
CurrentBest := Solution
while not StoppingCriteria do
    for m := 1 to MarkovChainLength do
        NeighborSolution := NeighborhoodOperator(Solution)
        if NeighborSolution < Solution then
            if NeighborSolution < CurrentBest then
                CurrentBest := NeighborSolution
            end if
            Solution := NeighborSolution
        else
            if RandomNumber ≤ e^((CurrentSolution−NeighborSolution)/Temp) then
                Solution := NeighborSolution
            end if
        end if
    end for
    Temp := α * Temp
end while
Result := CurrentBest
```

4.3.1 Neighborhood operator

The neighborhood operator adjusts the current solution. Since the constructive heuristic does not make trade-offs between increasing capacity or allowing tardiness, the neighborhood operator adjusts either a tardy order or an order using overtime. Which order uses overtime is an arbitrary choice. It depends on the order planned at the moment that regular capacity is not available anymore.

At each iteration of the improvement heuristic, a random order i is selected from the created set. If the randomly selected order is tardy, the logic in paragraph 1A is followed. If the randomly selected order uses overtime, the logic in paragraph 1B is followed. For both situations, precedence constraints may be violated. To fix violations of the precedence constraints, the logic in paragraph 2 is followed.

1A. Replanning tardy order. For a tardy order, we check if we can finish the last operation z_i in an earlier time bucket while using overtime to avoid tardiness. So, we check if the tardy order can be planned in an ATW window starting from the completion time of the last operation z_i minus 1 until the minimum time bucket in the planning horizon, i.e. $[ct_i - 1, T_{min}]$. After setting the ATW window, the last operation z_i is replanned using the logic in Figure 4.3.

1B. Replanning orders using overtime. If the order uses overtime, a random operation using overtime is selected. We check if some or all amount of overtime used can be planned in regular time in $t - 1$ or $t + 1$. If that is possible, we plan the available amount in regular time in the time bucket that has the most regular capacity available. If it is not possible, the next operation in order i using overtime is checked.

It may happen that no regular time is available for the operations using overtime in order i. To still find a neighbor solution, the order is replanned using an ATW window using the completion time of the last operation z_i. If the completion time of z_i (indicated by ct_i) is greater or equal to the minimum time bucket and less than the maximum time bucket, the ATW window is set to $[ct_i + 1, T_{min}]$. If the completion time of z_i is equal to the maximum time bucket in the planning horizon, the ATW window is set to $[ct_i - 1, T_{min}]$.

2. Violation of precedence constraints. Replanning the last operation of a tardy order or an operation using overtime may lead to a violation of the precedence constraints in the following ways:

1. The operation (except for the first operation) is replanned earlier than its predecessor finishes.
2. The operation (except for the last operation) is replanned later than its successor starts.

If precedence constraints are violated after replanning, all operations causing the violations are replanned until no violation exists. Replanning the operations due to precedence violation uses the logic in Figure 4.3.

Figure 4.4 shows a graphical representation of the neighborhood operator.

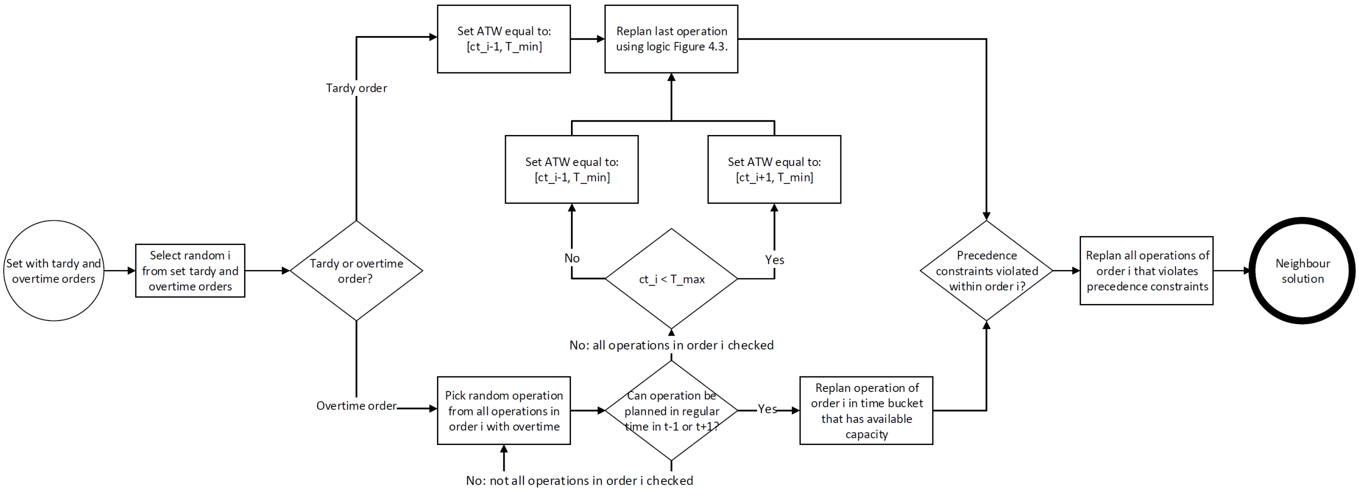

Figure 4.4: Flowchart of the neighborhood operator.

4.4 Conclusion on solution design

This chapter proposes a resource loading model formulation for the internal due date setting and the tactical production planning problem of HA. The model takes into account non-preemption between an operation of an order (i.e. between S_{is}), regular and overtime capacity constraints, and precedence constraints. We use the MIP model to plan a given set of orders in a specific planning horizon. In that way, we can evaluate if the Customer Service Level of HA can increase and against which costs.

To solve larger data instances, we designed a simple constructive and improvement heuristic. The constructive heuristic is designed to construct an initial solution. The improvement heuristic improves the initial constructed solution. Since the constructive heuristic does not make decisions on tardiness, we improve the initial solution by either replanning a tardy order or replanning an order using overtime.

The constructive heuristic is inspired by PBFL. It uses priority rules to determine the sequence in which orders are planned. For each S_{is} it sets an ATW window and plans S_{is} in the ATW window in one or multiple consecutive time buckets. If not enough regular capacity or overtime is available in the ATW window, we enlarge the ATW window. In the end, the objective value is calculated. One limitation is that the constructive heuristic only calculates tardiness and does not make decisions on tardiness. The improvement heuristic replans a tardy order or an order using overtime. In the first case, we check if a tardy order can be planned in an ATW window from the completion time of the last operation z_i minus 1. So, decreasing the ATW window. In the second case, we check if an operation of an order using overtime can be planned in regular time in $t-1$ or $t+1$. For both cases, we check if precedence constraints are violated after replanning. If they are violated, we replan the operations causing the violations until no violations exist.

5 Solution test

This chapter evaluates the performance of the solution design suggested in Chapter 4. Section 5.1 describes the setup of the solution test. Section 5.2 describes the experiments on the designed MIP model. Section 5.3 describes the experiments for the designed constructive heuristic. Section 5.4 describes the experiments for the designed improvement heuristic. Section 5.5 evaluates the performance of the constructive and improvement heuristic against the MIP model. Section 5.6 concludes the solution test.

5.1 Solution test setup

Section 5.1.1 describes the experiments of this research together with their goals. Section 5.1.2 describes the parameters and the values used in the experiment. Section 5.1.3 describes the data set.

5.1.1 Experiments and their goals

In this chapter, several experiments are conducted for the MIP model, constructive heuristic and improvement heuristic. The following experiments with their goals are conducted:

1. **MIP: impact of planning horizon length & workload** The goal is to identify and provide insights into the performance of the MIP for different planning horizon lengths and workloads.
2. **MIP: impact of different maximum overtime.** The goal is to identify the impact of using different maximum overtime and to provide insights into the maximum overtime needed to create feasible solutions.
3. **MIP: impact of different ϵ.** ϵ indicates the minimum duration required to start an operation. The goal of this experiment is to identify the impact of different ϵ on the performance of the MIP model.
4. **MIP: insight in Customer Service Level.** The goal is to provide insights into the extent the Customer Service Level can be increased against which costs. This test provides the following insights to HA: (1) the extent to which the Customer Service Level improves against which costs, (2) the bottleneck operations, and (3) the maximum overtime used to achieve the result.
5. **MIP: validation of infeasible production plans.** The MIP model results in two infeasible scenarios at the Customer Service Level tests. The goal of this test is to validate if the infeasible production plans resulting from the MIP model also appeared to be infeasible in practice.
6. **Constructive heuristic: impact of different priority rules.** The goal is to identify which priority rule works best for the situation of HA.
7. **Improvement heuristic: initialisation of the parameters.** The goal is to identify the initial values of the SA parameters that work best for the situation of HA. The parameters are initial temperature, length of Markov Chain, cooling scheme and stopping criteria.
8. **Solution quality of constructive heuristic and improvement heuristic.** The goal is to provide insights into the performance of the constructive and improvement heuristic.

5.1.2 Parameter settings

The following parameter values are retrieved from Isah and fixed during the experiments: S_{is}, p_{is}, d_i and c_j. The maximum overtime parameter mo_{jt} is determined in the experiments since they differ for each experiment. The cost of tardiness and overtime are equal for each experiment. The costs are determined by the management of HA. The cost of overtime is set to the gross wage per hour of an employee plus the overhead cost for equipment that they use. The cost of a tardy order is based on the risk of harming a customer relationship due to late delivery in combination with the indirect personnel cost of organising a late shipment. HA uses a cost of tardiness independent of the order size. HA does not distinguish between the importance of the relationship of customers with higher sales and organising a late shipment takes the same amount of time independent of the order sizes. The following costs are determined:

1. Cost of working one hour of overtime: €45
2. Cost of a tardy order: €500

5.1.3 Data

The data set used is retrieved from Isah. On 10-05-2023, we created a list of all production orders accepted to produce in 2023. This list includes orders already produced in 2023 and orders still to be produced in the remainder of 2023. A total of 582 orders and 2524 operations are accepted. Some operations in a production route have no processing time inserted in Isah. A processing time is defined for these operations based on the mean processing time of all operations in 2023. Table 5.1 shows the mean processing time (expressed in hours) per operation type j in 2023.

Table 5.1: Mean processing time per operation type j in 2023.

Operation	**Mean processing times (hours)**	**Operation**	**Mean processing times (hours)**
AOI	8	PWA	4
COAT	5	REW	1
EC	4	SMD	11
PB	1	SOL	2
POT	5	SSOL	9
PR	7	TC	1
PRNA	1	TE	5
PRO	4	VER	4
PW	3		

Two priority rules are based on the processing times of operations. Therefore some insight is gained into the distributions of the processing times of the operations in the data set. Figure 5.1 shows a graphical representation of the distribution of p_{is} in the data set of 2023. The dataset contains more small processing times (p_{is}) than larger processing times. This may lead to more tardiness if the larger orders are planned first since a few large orders late lead to less tardiness than a lot of small orders late. Therefore, we hypothesise that the priority rule SPT performs better than LPT. The processing times of zero in the Figure confirm that some operation in a production route have nu processing time inserted in Isah.

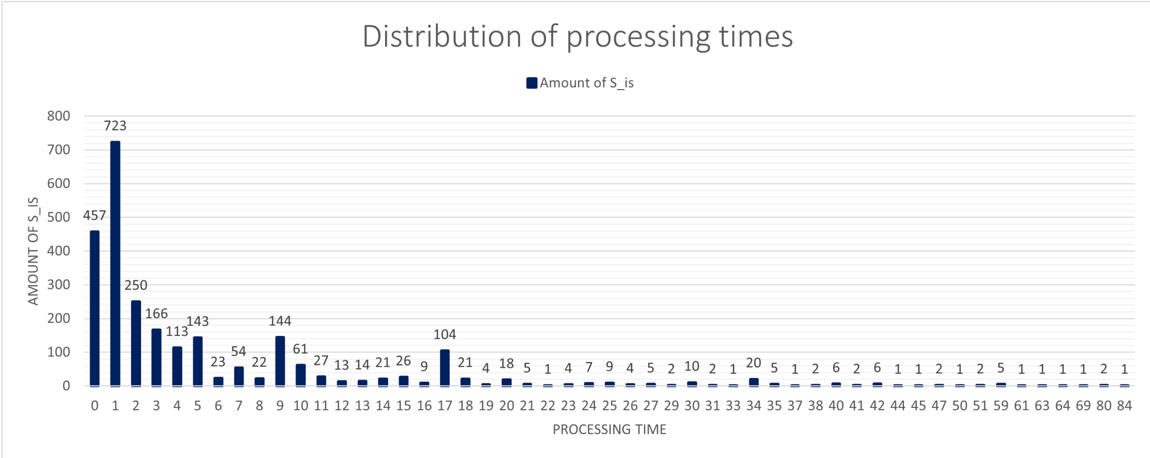

Figure 5.1: Distribution of processing times in dataset of 2023.

5.2 Experiments on MIP

Section 5.2.1 identifies the performance of the MIP model for different planning horizon lengths and workloads. Section 5.2.2 identifies the impact of different maximum overtimes on the MIP model. Section 5.2.3 identifies the impact of different minimum durations (ϵ) required to start an operation. Section 5.2.4 identifies and provides insights into the improvement of the Customer Service Level by using the MIP model. Section 5.2.5 validates the infeasibility found by the MIP model of the production plan.

5.2.1 MIP: impact of planning horizon length & workload

This section identifies and provides insights into the performance for different planning horizon lengths and workload sizes. We assume no overlap between 2 consecutive planning horizons.

The operations manager of HA evaluates and updates the production plan every 2 months to create a more reliable and feasible production plan based on the current situation. The first planning horizon is set to 2 months (8 time buckets). Also, a smaller planning horizon of 1 month (4 time buckets) and two larger planning horizons of 4 months (16 time buckets) and 6 months (24 time buckets) are tested. HA knows a light, typical and heavy workload. The workload at HA depends on the processing time and not the number of orders, i.e. the workload is the sum of all processing times ($\sum_{is} p_{is}$). So it may happen that a heavy workload contains more orders than a typical workload. Each workload type is based on the workload per month in the planning horizon, i.e. $\sum_{is} p_{is}$ divided by the planning horizon length. Table 5.2 provides the thresholds for each workload type. We test a light workload period, a typical workload period, and a heavy workload period for each planning horizon.

Table 5.2: Definition of workload types at HA.

Workload type	Average workload per month in planning horizon (hours)
Light	< 1000
Typical	1000-1800
Heavy	> 1800

We identify scenarios for each planning horizon length and workload from the dataset of 2023. Table 5.3 shows the resulting scenarios.

Table 5.3: Scenarios test MIP: impact of planning horizon length & workload.

Scenario	Planning horizon (time buckets)	Workload	Time buckets	# orders	# operations	Workload (hours)
1	4	Light	26-29	33	150	818
2	4	Typical	11-14	75	298	1714
3	4	Heavy	7-10	69	279	1921
4	8	Light	26-33	50	226	1415
5	8	Typical	23-30	94	423	2456
6	8	Heavy	6-13	148	604	3825
7	16	Light	26-41	91	441	2768
8	16	Typical	19-34	177	780	4989
9	16	Heavy	6-21	289	1201	8007
10	24	Light	25-48	130	625	4209
11	24	Typical	16-39	261	1171	7759
12	24	Heavy	2-25	430	1798	11786

We start the experiments with a maximum overtime of 60 for each scenario test. If maximum overtime of 60 is not enough to create a feasible solution or a solution with a gap of 0%, we test a higher maximum amount of overtime. We stop the MIP if a solving time of 600 seconds is reached since HA does not accept a solution solving for more than 10 minutes.

Table 5.4 shows the results of the experiments. A combination of the planning horizon length and workload determines if the MIP is efficient to use within a reasonable amount of time and if it can find an optimal solution. A solution is optimal if the objective value found equals the lower bound of the minimisation problem. The lower bound of a minimisation problem represents the best feasible solution value for the objective function within its feasible region (Lalla, 2021). The upper bound of a minimisation problem represents the best feasible solution found so far (Lalla, 2021). A gap represents the difference between the lower bound and the upper bound. A gap of 0% demonstrates optimality. The result also includes bottleneck operations per scenario. Bottleneck operations are operations using overtime.

For a planning horizon of 1 and 2 months, the MIP provides an optimal solution for a light, typical and heavy workload. For a planning horizon of 4 months, the MIP only provides an optimal solution for a light workload. The MIP model cannot find an optimal solution for a planning horizon of 6 months. For the planning horizons of 4 and 6 months, higher maximum overtime is tested to evaluate the performance of the MIP. For a planning horizon of 4 months with a typical workload, the MIP finds solutions with a gap of around 36%. For heavier workloads in combination with larger planning horizons, the MIP model is (1) infeasible or (2) has a large maximum overtime and a high solution gap which is far from optimal. The MIP tests in the remainder of this chapter are performed for a planning horizon of 1 and 2 months. The constructive and improvement heuristic may be more efficient for larger planning horizons, especially in combination with typical and heavy workloads. For all feasible scenarios, SSOL is a bottleneck operation. Other bottleneck operations are SMD, PR and AOI.

Table 5.4: Result test MIP: impact of planning horizon length & workload.

Scenario	Maximum overtime	Cost of tardiness (€)	Cost of overtime (€)	Total cost (€)	CPU time	GAP (%)	Bottleneck operations
1	60	0	0	0	0.1	0	-
2	60	2000	13860	15860	0.8	0	SMD,SSOL
3	63	1000	21105	22105	13	0	SMD,SSOL
4	60	0	0	0	1	0	-
5	60	2000	1575	3575	19	0	SSOL
6	60	1500	49500	51000	148	0	SMD,SSOL
7	60	0	0	0	30	0	-
8a	60	7000	5985	12985	$\geq$ 600	38	SSOL,PR
8b	180	7000	4590	11590	$\geq$ 600	29	SSOL
8c	240	7500	6750	14250	$\geq$ 600	43	SSOL
9a	60	-	-	-	$\geq$ 600	infeasible	-
9b	240	-	-	-	$\geq$ 600	infeasible	-
9c	999	2500	113625	116125	$\geq$ 600	5	SMD,SSOL,PR
10	60	1000	0	1000	156	0	-
11a	60	-	-	-	$\geq$ 600	infeasible	-
11b	240	-	-	-	$\geq$ 600	infeasible	-
11c	999	106000	76185	182185	$\geq$ 600	92	SSOL,AOI
12a	60	-	-	-	$\geq$ 600	infeasible	-
12b	240	-	-	-	$\geq$ 600	infeasible	-
12c	999	2000	162315	164315	$\geq$ 600	81	SMD,SSOL,PR

5.2.2 MIP: impact of different maximum overtime

Section 5.2.1 concludes that the MIP model leads to optimal solutions for a planning horizon of 1 and 2 months for each workload type. Therefore, the impact of maximum overtime is tested for a planning horizon of 1 and 2 months in combination with light, typical and heavy workloads. These are scenarios 1, 2, 3, 4, 5 and 6 of Table 5.3 in Section 5.2.1. We also provide insights into the maximum overtime needed to create feasible solutions.

Table 5.5 shows the results of the experiments. We find the maximum overtime needed to create a feasible solution for each scenario. The solution improves a little by increasing overtime. This is logical since increasing available overtime enables orders to use overtime instead of being tardy. The solution cannot increase tardiness to improve the objective value because all frequently occurring operations use all regular capacity available in each time bucket in the planning horizon. This implies that we cannot improve the solution by finishing orders late because no regular capacity is available in the later time buckets. The impact of maximum overtime is low. By increasing overtime, we only can decrease tardiness by 1 or 2 orders even if overtime is increased significantly.

Table 5.5: Result test MIP: impact of maximum overtime.

Scenario	**Maximum overtime**	**Tardy orders (%)**	**Cost of tardiness (€)**	**Cost of overtime (€)**	**Total cost (€)**	**CPU time**	**Bottleneck operations**
1	0	0	0	0	0	0.1	-
2a	42	-	-	-	Infeasible	0.8	-
2b	43	5.3	2000	13860	15860	1	SMD,SSOL
2c	60	5.3	2000	13860	15860	1	SMD,SSOL
2d	100	4	1500	14265	15765	0.8	SMD,SSOL
2e	400	4	1500	14265	15765	0.8	SMD,SSOL
3a	62		-	-	Infeasible	0.1	-
3b	63	2.9	1000	21105	22105	14	SMD,SSOL
3c	100	1.4	500	21105	21605	25	SMD,SSOL
4	0	0	0	0	0	1	-
5a	4	-	-	-	Infeasible	0.1	-
5b	5	5.3	2500	1575	4075	42	SSOL
5c	25	4.3	2000	1575	3575	19	SSOL
5d	60	4.3	2000	1575	3575	18	SSOL
5e	400	4.3	2000	1575	3575	17	SSOL
6a	57	-	-	-	Infeasible	1	-
6b	58	2.0	1500	49500	51000	299	SMD,SSOL,PR
6c	60	2.0	1500	49500	51000	153	SMD,SSOL,PR
6d	100	1.4	1000	49500	50500	157	SMD,SSOL,PR
6e	400	1.4	1000	49500	51000	$\geq$ 600	SMD,SSOL,PR

5.2.3 MIP: impact of different ϵ

Again, we use scenarios 1, 2, 3, 4, 5 and 6 of Table 5.3 in Section 5.2.1. Section 5.2.2 concludes that the impact of maximum overtime on the MIP model is low. For this section, we use the maximum overtime found such that the scenario is feasible. Table 5.6 shows the resulting scenarios.

Table 5.6: Scenarios test MIP: impact of different ϵ.

Scenario	**Planning horizon (time buckets)**	**Workload**	**Maximum overtime**	**Time buckets**	**# orders**	**# operations**	**Workload (hours)**
1	4	Light	0	26-29	33	150	818
2	4	Typical	43	11-14	75	298	1714
3	4	Heavy	63	7-10	69	279	1921
4	8	Light	0	26-33	50	226	1415
5	8	Typical	5	23-30	94	423	2456
6	8	Heavy	58	6-13	148	604	3825

This section identifies the impact of different ϵ, i.e. the minimum duration required to start an operation. We test the following minimum durations: 0.01, 0.1, 0.25, 0.5 and 0.75. We do not test an ϵ greater than 1 since this leads to infeasible solutions. This is logical since a lot of processing times equals 1 (see Figure 5.1). We do not have processing times lower than 1.

Table 5.7 shows the results of the experiments. We can conclude that the epsilon does not have an influence on the performance of the MIP. Even the solving time remains almost the same for each epsilon. This implies that it is not more efficient to use a minimum duration lower than 1.

Table 5.7: Result test MIP: impact of different ϵ.

Scenario	ϵ	**Cost of tardiness (€)**	**Cost of overtime (€)**	**Total cost (€)**	**CPU time**	**Bottleneck operations**
1	0.01, 0.1, 0.25 0.5, 0.75	0	0	0	± 0.1	-
2	0.01, 0.1, 0.25 0.5, 0.75	2000	13860	15860	± 1	SMD,SSOL
3	0.01, 0.1, 0.25 0.5, 0.75	1000	21105	22105	± 33	SMD,SSOL,AOI
4	0.01, 0.1, 0.25 0.5, 0.75	0	0	0	± 1	-
5	0.01, 0.1, 0.25 0.5, 0.75	2500	1575	4075	± 21	SSOL
6	0.01, 0.1, 0.25 0.5, 0.75	1500	49500	51000	± 435	SMD,SSOL,PR

5.2.4 MIP: insight in Customer Service Level

The previous sections analyse the impact of different settings of the MIP model. The goal of analysing the improvement of the Customer Service Level is to improve the Customer Service Level against the lowest cost. We use the conclusions of the previous tests to set the settings of the MIP model. We again use scenarios 1, 2, 3, 4, 5 and 6 of Table 5.3 in Section 5.2.1. For the maximum overtime, we test two cases: (1) no restriction on maximum overtime and (2) the current maximum overtime allowed at HA (i.e. $mo_{jt} = 20$). The first case is used to find a cost-efficient balance between overtime and tardiness without maximum overtime restrictions. The second case is used to identify if the scenarios are feasible with the current maximum overtime restrictions of HA. The first case is indicated with scenario + a, whereas the second case is indicated with scenario + b. For both cases, we use an ϵ of 1. Table 5.8 shows the results of the experiments. The last column indicates the maximum overtime actually used at the solution.

We conclude that the maximum overtime of HA of 20 only provides a feasible solution for a planning horizon of 1 and 2 months in combination with a light workload. For these combinations, also 0% tardiness is achieved. The planning horizons of 1 and 2 months with a light, typical and heavy workload are feasible when there is no restriction on maximum overtime. When using no maximum overtime restrictions, we still have a small percentage of tardy orders. The goal of HA is a Customer Service Level of at least 90%. If HA is able to extend the capacity to the maximum overtime used against the cost mentioned, HA is able to improve its Customer Service Level above 90%. The bottleneck for each scenario is SMD and SSOL. Also, AOI and PR are bottlenecks in some scenarios.

Table 5.8: Result test MIP: Customer Service Level.

Scenario	Tardy orders (%)	Cost of tardiness (€)	Cost of overtime (€)	Total cost (€)	Bottleneck operations	Max. overtime used
1a	0	0	0	0	-	0
1b	0	0	0	0	-	0
2a	4.0	1500	14264	15765	SMD,SSOL,AOI	74
2b	-	-	-	infeasible	-	-
3a	1.4	500	21105	21605	SMD,SSOl,AOI	238
3b	-	-	-	infeasible	-	-
4a	0	0	0	0	-	0
4b	0	0	0	0	-	0
5a	4.3	2000	1575	3575	SSOL	18
5b	4.3	2000	1575	3575	SSOL	14
6a	1.4	1000	49500	50500	PR,SMD,SSOL	127
6b	-	-	-	infeasible	-	-

5.2.5 MIP: validation of infeasible production plans

The test on Customer Service Level in Section 5.2.4 shows two infeasible scenarios: a planning horizon of 4 time buckets with a heavy workload (scenario 3b) and a planning horizon of 8 time buckets with a heavy workload (scenario 6b). Both scenarios use the current maximum overtime at HA of 20 hours per operation per time bucket. We test for both scenarios if they appeared infeasible in practice. Section 4.1.1 discusses that a solution is feasible if each operation is completely assigned within the planning horizon while respecting the precedence, capacity and non-preemption constraints. To validate the infeasibility of the model, we examine whether each order within the planning horizon is completed by or before its internal due date. Orders finishing beyond the planning horizon demonstrate the infeasibility of the scenario.

Table 5.9 shows the result in terms of orders finished outside the planning horizon and the number of tardy orders. The production plan in both scenarios is infeasible in practice due to orders finished outside the planning horizon. A significant amount of orders are tardy. We analysed the number of orders finished beyond the planning horizon between scenario 3b and scenario 6b. One potential explanation is that scenario 6b has a broader planning horizon, allowing for a more distributed allocation of orders across the time buckets. Orders with internal due dates in time buckets 7 to 10 fall within the planning horizon due to the inclusion of time buckets 6, 11, 12 and 13 in the planning horizon in scenario 6b. However, the orders from scenario 3b planned in time buckets 6, 11, 12 and 13 limit the capacity available for the orders with an internal due date in those time buckets and they are planned outside the planning horizon of 6-13. Besides, there is an almost equal amount of due orders in each time bucket each involving comparable numbers of operations and processing times that need to be planned. So, this explanation seems unlikely. The number of tardy orders in each scenario is almost equal.

The infeasibility in practice could have been avoided by the MIP model which already showed the infeasibility of the problem by a maximum overtime of 20 hours. By using the MIP model, HA could analyse the situation and take preventive measures, including the adjustment of the maximum overtime or proactive communication with customers regarding revised shipment dates. Implementing these proactive steps ensures that the problem becomes feasible and reduces the number of orders that are delayed.

Table 5.9: Result test MIP: validation on infeasible production plans.

Scenario	**Planning horizon (time buckets)**	**# orders**	**Orders finished outside planning horizon (%)**	**Tardy orders (%)**
3b	7-10	69	70	77
6b	6-13	148	51	74

5.3 Experiment on the constructive heuristic: priority rules

This section identifies which priority rule leads to the lowest objective. The priority rules mentioned in Section 4.2.1 are EDD, LDD, SPT and LST. We test the priority rules for a planning horizon of 2 months for each workload type since the standard planning horizon of HA is two months. We use the maximum overtime found for a feasible solution in Section 5.2.2. If for the constructive heuristic the problem is infeasible using the maximum overtime of Section 5.2.3 Table 5.6, we increase the maximum overtime until the constructive problem is feasible. We compare two results:

1. The maximum overtime needed for a scenario and specific priority rule to be feasible.
2. The performance of the different priority rules in a scenario for the same maximum overtime.

Table 5.10 shows the result of the first experiment. The SPT priority rule needs the highest maximum overtime for a typical (scenario 5) and heavy (scenario 6) workload to be feasible. The SPT priority rule also leads to the lowest percentage of tardy orders because more overtime can be used such that fewer orders are tardy.

Table 5.10: Performance of constructive heuristic for different priority rules: what maximum overtime needed?

Scenario	**Maximum overtime**	**Priority rule**	**Tardy orders (%)**	**Cost of tardiness (€)**	**Cost of overtime (€)**	**Total cost (€)**
4	0	EDD	6.0	1500	0	1500
4	0	LDD	8.0	2000	0	2000
4	0	SPT	6.0	1500	0	1500
4	0	LPT	6.0	1500	0	1500
5	5	EDD	41.5	19500	4860	24360
5	11	LDD	10.6	5000	11610	16610
5	22	SPT	4.3	2000	14265	16265
5	7	LPT	58.5	27500	8145	35645
6	58	EDD	16.9	12500	62595	75095
6	59	LDD	7.4	5500	60885	66385
6	71	SPT	1.4	1000	61515	62515
6	58	LPT	17.6	13000	62505	75505

Table 5.11 shows the result of the second experiment. The priority rule SPT leads to the lowest objective value for a typical (scenario 5) and heavy (scenario 6) workload. For scenario 6, SPT also leads to the lowest percentage of tardy orders. For scenario 5, SPT and LDD both lead to the lowest percentage of tardy orders.

Table 5.11: Performance of constructive heuristic for different priority rules using same amount of maximum overtime.

Scenario	**Maximum overtime**	**Priority rule**	**Tardy orders (%)**	**Cost of tardiness (€)**	**Cost of overtime (€)**	**Total cost (€)**
4	0	EDD	6.0	1500	0	1500
4	0	LDD	8.0	2000	0	2000
4	0	SPT	6.0	1500	0	1500
4	0	LPT	6.0	1500	0	1500
5	22	EDD	12.8	6000	16335	22335
5	22	LDD	4.3	2000	15660	17660
5	22	SPT	4.3	2000	14265	16265
5	22	LPT	45.7	21500	18315	39815
6	71	EDD	4.1	3000	62460	65460
6	71	LDD	2.7	2000	61560	63560
6	71	SPT	1.4	1000	61515	62515
6	71	LPT	6.1	4500	63450	67950

5.4 Experiments on improvement heuristic: initialisation parameters

For SA, the following parameters need initialisation: initial temperature, length of Markov Chains, cooling scheme and stopping condition. Each parameter is set by experiments. We use scenario 5 (planning horizon of 2 months with a typical workload) to find the initial parameters since this is the current planning horizon and preferred workload type for HA. The SPT priority rule is used to construct a solution since this is the best-performing priority rule for our problem according to Section 5.3.

5.4.1 Initial temperature

Section 3.2.2 describes that the initial temperature is set such that the initial acceptance ratio is approximately 1. At this stage, the Markov Chain length, cooling scheme and stopping condition are not determined yet. For the Markov Chain length, a rule of thumb is to use the number of neighbour solutions. For this experiment, we use a smaller and larger arbitrary Markov Chain length to test the influence of the Markov Chain length on the initial temperature. The smaller Markov Chain length is set to 100, the bigger is set to 1000. For the cooling scheme, we choose α close to 1. We choose an α of 0.99. For the stopping criteria, we use a common stopping criterion where the heuristic stops if the temperature is getting close to 0. During this experiment, we stop if the temperature drops below 0.05. We start with an initial temperature of 10 and increase it until the initial acceptance ratio becomes around 1. Figure 5.2 shows the results of the experiments with a Markov Chain length of 100 and Figure 5.3 shows the results of the experiments with a Markov Chain length of 1000.

The Markov Chain length has almost no effect on the acceptance ratio for different initial temperatures while the CPU time increases significantly with a larger Markov Chain length. The CPU time for one iteration of Markov Chain length 100 is around 4, whereas the CPU time for one iteration of Markov Chain length 1000 is around 180 ($\pm$ 3 minutes). The number of iterations needed for an initial temperature is calculated by $initial\ temperature * \alpha^n = stopping\ condition$ where n equals the number of iterations needed. In this experiment, n is calculated by $n = \log_{0.99}(\frac{0.05}{Initial\ temperature})$. Note that the number of iterations highly depends on α, so the number of iterations found in this experiment only applies to $\alpha = 0.99$ and a stopping condition of 0.05.

A higher initial temperature implies more iterations, so we make a trade-off between the acceptance ratio and the number of iterations needed. For both Markov Chain lengths, the initial acceptance ratio is above 0.95 for an initial temperature of 3000. So, we choose an initial temperature of 300. We do not choose a higher initial temperature since it increases the acceptance ratio by 0.01 while we need around 20 iterations more. For a Markov Chain length of 100, this is around 1.6 minutes. For a Markov Chain length of 1000, this is around 75 minutes.

Exerpiment nr.	Initial temperature	Accepted solutions	Proposed solutions	Acceptance ratio	Nr. iterations until stopping criterion reached
1	10	35	93	0,38	527
2	100	42	94	0,45	756
3	1000	63	74	0,85	985
4	1500	66	69	0,96	1026
5	2000	69	73	0,95	1054
6	2400	71	74	0,96	1072
7	3000	71	73	0,97	1095
8	4000	81	82	0,99	1123
9	5000	72	74	0,97	1146
10	7000	79	80	0,99	1179
11	9000	71	72	0,99	1204

Figure 5.2: Results initial temperature experiment with Markov Chain length of 100.

Exerpiment nr.	Initial temperature	Accepted solutions	Proposed solutions	Acceptance ratio	Nr. iterations until stopping criterion reached
1	10	380	962	0,40	527
2	100	540	973	0,55	756
3	1000	705	813	0,87	985
4	1500	722	777	0,93	1026
5	2000	688	743	0,93	1054
6	2400	743	785	0,95	1072
7	3000	705	736	0,96	1095
8	4000	735	755	0,97	1123
9	5000	738	753	0,98	1146
10	7000	746	754	0,99	1179
11	9000	759	769	0,99	1204

Figure 5.3: Results initial temperature experiment with Markov Chain length of 1000.

5.4.2 Markov Chain length, cooling scheme and stopping condition

With an initial temperature of 3000, we find the best Markov Chain length and cooling scheme. The trade-off is a good solution within a reasonable amount of time. Section 5.4.1 explains how to calculate the number of iterations needed given an initial temperature, α and stopping condition. We first emphasise the influence of α on the number of iterations. Assume a stopping condition of Temperature < 0.05 and an initial temperature of 3000. Table 5.12 shows the number of iterations needed for some values of α. Especially for α close to 1, the number of iterations increases significantly. We start testing the following values of α: 0.85, 0.90, 0.96 and 0.98. Based on the results different α is tested. We start the experiments with a Markov Chain length of 100 and will increase and decrease according to the results.

Table 5.12: Impact of α on the number of iterations needed for SA.

α	**Number of iterations needed**	α	**Number of iterations needed**
0.80	49	0.96	270
0.85	68	0.97	361
0.90	104	0.98	545
0.95	215	0.99	1095

Figure 5.4 shows the results of the experiments. The SA heuristic is almost insensitive to the Markov chain Length and decrease factor. This implies that the current best solution is found quickly and the heuristic is not able to improve the solution more. This indicates that the SA neighborhood operator can be designed in a better way. Due to time limitations of this research and the sufficiency of the MIP model for HA, we are not improving the neighborhood operator. HA commonly uses a planning horizon of around 2 months, which is solvable by the MIP model. Improving the SA heuristic is a recommendation for future research (see Section 6.2). The improvement heuristic improves the constructive solutions on average with 19%. For the remainder of the experiments, we use a Markov Chain length of 25 and an α of 0.90 since this combination leads to the lowest CPU time against the best improvement to the constructive solution.

Experiment nr.	Markov Chain Length	Decrease factor	Tardy orders	Cost of tardiness	Cost of overtime	Total cost	CPU time	GAP MIP	Improvement to constructive
1	100	0.85	4,3%	€ 2.000	€ 12.330	€ 14.330	232	301%	19%
2	100	0.90	3,2%	€ 1.500	€ 12.465	€ 13.965	355	291%	21%
3	100	0.96	4,3%	€ 2.000	€ 13.635	€ 15.635	933	337%	12%
4	100	0.98	5,3%	€ 2.500	€ 13.500	€ 16.000	1877	348%	10%
5	150	0.85	3,2%	€ 1.500	€ 12.465	€ 13.965	343	291%	21%
6	150	0.9	4,3%	€ 2.000	€ 12.330	€ 14.330	532	301%	19%
7	50	0.85	5,3%	€ 2.500	€ 12.240	€ 14.740	118	312%	17%
8	50	0.90	5,3%	€ 2.500	€ 13.500	€ 16.000	183	348%	10%
9	50	0.96	4,3%	€ 2.000	€ 12.330	€ 14.330	467	301%	19%
10	50	0.98	4,3%	€ 2.000	€ 12.330	€ 14.330	923	301%	19%
11	25	0.85	4,3%	€ 2.000	€ 12.330	€ 14.330	57	301%	19%
12	25	0.90	3,2%	€ 1.500	€ 12.465	€ 13.965	89	291%	21%
13	25	0.96	4,3%	€ 2.000	€ 12.330	€ 14.330	236	301%	19%

Figure 5.4: Results of the Markov Chain length and cooling scheme experiments.

Since the heuristic finds the current best solution quickly, we assume that the stopping condition does not have a high influence as well. An experiment testing different stopping conditions confirms our assumptions. The solution quality does not improve using a lower or higher stopping condition. Therefore, we use the stopping condition used during the previous experiments: the temperature drops below 0.05.

5.5 Solution quality of constructive and improvement heuristic

Despite the fact that the neighborhood operator can be designed in a better way, the solution quality of the constructive and improvement heuristic is compared to the MIP solutions. For the constructive heuristic, we use the priority rule SPT. For SA, we use the parameter values found in Section 5.4. So an initial temperature of 3000, a Markov Chain length of 25, an α of 0.90 and a stopping condition of temperature below 0.05. We test scenarios 1, 2, 3, 4, 5 and 6 of Section 5.2.1 Table 5.3. We do not test the other scenarios because we cannot compare an optimal MIP solution with the constructive and improvement solutions since the MIP model cannot find an optimal solution for these scenarios.

Table 5.13 shows the results. In terms of tardiness, the constructive and improvement heuristic performs better than the MIP model. The constructive heuristic results in no tardiness. Section 4.2.3 discusses that the constructive heuristic does not make decisions on tardiness or using overtime. It plans the order in (consecutive) time bucket(s) closest to the internal due date. Since there is no maximum overtime restriction, overtime is increased until each order is finished on its internal due date. Therefore, the constructive heuristic results in no tardiness against a higher cost for using overtime. The improvement heuristic slightly increases the percentage of tardy orders since orders using overtime are rescheduled which leads to a tardy order. In terms of objective value, the MIP performs better than the constructive and improvement heuristic. The MIP makes decisions on tardiness and using overtime which leads to a cost-efficient solution. This confirms the assumption that the neighborhood operator can be designed in a better way such that the amount of overtime and tardiness leads to a more cost-efficient solution.

Table 5.13: Performance of the three models: tardy orders and objective values.

Scenario	**Tardy orders MIP (%)**	**Total cost MIP (€)**	**Tardy orders constructive (%)**	**Total cost constructive (€)**	**Tardy orders SA (%)**	**Total cost SA (€)**
1	0	0	0	4545	0	360
2	4.0	15765	0	21150	2.7	21065
3	1.4	21605	0	29205	0	26820
4	0	0	0	5715	1.8	1040
5	4.3	3575	0	21195	2.0	16525
6	1.4	50500	0	65745	2.7	62615

Table 5.14 provides insights into the gap between the MIP and constructive heuristic, MIP and improvement heuristic, as well as the improvement of the constructive heuristic by the improvement heuristic. In scenarios 2, 3 and 6 the constructive and improvement heuristic demonstrate a performance with gaps ranging from 24% to 35.2%. There are two already known explanations for the gaps. The first explanation is the SA neighborhood operator. It finds the current best solution quickly and is not able to improve the solution close to the MIP solution. The second explanation is

the constructive heuristic. The constructive heuristic does not make decisions on tardiness or using overtime. It plans the operation of an order backwards in the set ATW window and changes the ATW window if not enough capacity or overtime is available. Since there is no maximum overtime restriction, the operations of the order are always scheduled in the first time bucket of the ATW window (i.e. the internal due date) using overtime. To decrease overtime cost, the improvement heuristic reschedules the orders using overtime (partially) to an earlier or later time bucket such that overtime cost decreases.

A contradicting insight resulting from this test is the improvement of the constructive heuristic by SA of 92.1% (scenario 1) and 81.8% (scenario 4). The MIP model shows that scenario 1 and 4 is solvable without tardiness and overtime by scheduling some operation in consecutive time buckets instead of one time bucket. Since the improvement heuristic reschedules some operations (partially) to an earlier or later time bucket and the scenarios have a light workload, the improvement heuristic is able to improve the constructive heuristic significantly. The second notable insight is scenario 5 with a gap between the MIP model and constructive heuristic of 492% and the gap between the MIP model and improvement heuristic of 362%. This implies that the MIP model finds a significantly better solution compared to the constructive and improvement heuristic. The main difference between the MIP and the heuristics is the amount of overtime used. The MIP only uses 35 hours of overtime (€1575), whereas the constructive heuristic uses 471 hours of overtime (€21195) and the improvement heuristic uses 345 hours of overtime (€15525). An explanation is that scenario 5 contains orders with high processing time operations around the same internal due date. Since there is no maximum overtime restriction, the constructive heuristic increases the capacity to fit the high-processing time operations into the time bucket of the internal due date whereas the MIP model balances the high-processing times over multiple time buckets. Because of the high gap between the MIP and constructive, the improvement heuristic is able to improve the constructed solution significantly compared to the improvements in scenarios 2, 3 and 6. However, it cannot improve close to the MIP solution due to the SA neighborhood operator.

Table 5.14: Performance of the three models: the gap between the solutions.

Scenario	**Gap between MIP & constructive (%)**	**Gap between MIP & SA (%)**	**Improvement constructive by SA (%)**
1	-	-	92.1
2	34.2	33.6	0.4
3	35.2	24.1	8.2
4	-	-	81.8
5	492.9	362.0	22.0
6	30.2	24.0	4.8

5.6 Conclusion on solution test

In this chapter, we use scenarios representing a planning horizon of 1, 2, 4 or 6 months in combination with a light, typical or heavy workload. The scenarios contain some undefined processing times. The undefined processing times are defined based on historical data. The orders contain more operations with smaller than larger processing times.

The MIP model is computationally tractable for planning horizons of 1 and 2 months with a light, typical and heavy workload. The impact of the maximum overtime and ϵ in the MIP model are low. The Customer Service Level of HA can improve beyond 90% at the expense of overtime and tardiness costs when following the production plan provided by the MIP model. The bottleneck operations are SMD, SSOL, AOI and PR.

For planning horizons larger than 2 months with a typical or heavy workload, the MIP model becomes computationally expensive. For these situations, constructive and improvement heuristics are used. The best priority rule for the constructive heuristic is the Shortest Processing Time. For the improvement heuristic, the initial parameter values for the temperature (3000), Markov Chain length (25), decrease factor (0.9) and stopping condition (temperature < 0.05) are defined. The improvement heuristic is almost insensitive for the Markov Chain length, decrease factor and stopping condition. For almost all scenarios, the constructive and improvement heuristic performs with a gap of 24% and 35.2% between the MIP solution. This is a consequence of the SA neighborhood operator and constructive heuristic. The SA neighborhood operator finds the best solution quickly but is not able to improve the solution close to the MIP solution. The constructive heuristic does not make decisions on tardiness and overtime. Without overtime restrictions, the constructive heuristic plans each operation at the internal due date of the order by increasing overtime. It does not create a tardy order if that would lead to lower total costs. Due to time limitations, this is not executed during this research but is a recommendation for future research (see Section 6.2).

6 Conclusion and recommendation

Section 6.1 provides a conclusion to this research and Section 6.2 provides recommendations for HA. Section 6.3 discusses the contribution of this research to theory and practice. Section 6.4 discusses implementation challenges for HA.

6.1 Conclusion

HA employs a high-mix low-volume MTO production strategy in a job shop production environment. The management of HA finds the Customer Service Level of 86.6% too low and wants to improve the Customer Service Level to at least 90%. After identifying possible (core) problems, we conclude that the low Customer Service Level is a result of end products missing their external due date for customer shipments. The external due dates are missed due to infeasible production plans that rely on assumptions about resource capacity instead of incorporating actual capacity data. Therefore, the main research question is:

How can a method that takes into account resource capacity using data available in tactical production planning for a high-mix low-volume make-to-order EMS company be designed such that the Customer Service Level improves from 86.6% to 90%?

After limiting the scope, we conclude that resource loading is the most appropriate method to solve this problem. Resource loading measures the impact of a set of orders in terms of internal due dates and resource capacity levels by either allowing tardiness (resource-driven) or extending capacity (time-driven). The designed model for HA integrates both resource loading approaches. The designed MIP model schedules orders with its operations within the planning horizon while minimising tardiness and overtime costs. The MIP model allows both tardiness and overtime and takes into account non-preemption constraints within an operation of an order, capacity constraints and precedence constraints. The MIP model can be used by HA for a planning horizon of 1 or 2 months in combination with light, typical and heavy workloads. For other combinations of planning horizons and workloads, the MIP model becomes computationally expensive. The experiments throughout this research show that a Customer Service Level of at least 90% can be achieved by using the MIP against some tardiness and overtime costs. For the scenarios tested, the Customer Service Level can improve between 95% and 100% with costs ranging between €0 and €50,500.

For larger data instances a constructive and improvement heuristic is needed to solve the instance in a reasonable amount of time. For HA, larger data instances are a planning horizon larger than 2 months in combination with a typical and heavy workload. The constructive heuristic is based on Partial Backward Finite Loading. Partial Backward Finite Loading plans operations of an order in one or multiple contiguous time buckets with enough capacity left. Whereas Partial Backward Finite Loading takes into account capacity, some simpler priority rules also exist such as Earliest Due Date, Latest Due Date, Shortest Processing Time and Largest Processing Time. For HA, a constructive heuristic is designed based on Partial Backward Finite Loading while determining the order sequence by the Shortest Processing Time priority rule. To improve the constructive heuristic, an improvement heuristic is designed. The improvement heuristic is Simulated Annealing which adjusts the solution by either trying to schedule overtime in regular time or by decreasing the completion time of tardy orders. The constructive and improvement heuristics achieve a gap to the MIP solution between 24% and 35.2%. The are two reasons for the gaps. The first one is that the constructive heuristic does not decide on overtime and tardiness. It plans the operation of an order backwards in the set ATW window and only enlarges the ATW window if not enough capacity or overtime is available. Without maximum overtime restrictions, the constructive heuristic increases overtime until enough overtime is available to fit the operation at the internal due date of the order. So, it mainly increases overtime costs instead of creating tardiness costs. The second one is that the neighborhood operator can be designed in a better way. The experiments show that the SA heuristic is almost insensitive to the Markov Chain Length and decrease factor, so the current best solution is found quickly and the heuristic is not able to improve the solution more. Therefore, the SA heuristic cannot improve the constructed solution such that the objective value comes close to the MIP solution.

6.2 Recommendations

To goal of this research is to create a method that takes into account resource capacity using data available in tactical planning and to find out if the Customer Service Level can be improved beyond 90%. This section explains recommendations formed during this research.

Implement the MIP model at HA. We recommend using the designed model at the tactical planning level to gain insights into the resource capacity levels in a specific planning horizon and to set internal due dates such that the Customer Service Level can be improved. Section 6.4 discusses the implementation challenges for HA.

Further developing the constructive and improvement heuristic. The main focus of this research is the MIP model. A simple constructive heuristic and improvement heuristic are provided for larger data instances. For the constructive heuristic, we suggest researching the use of a greedy heuristic that makes decisions on tardiness versus overtime. For the improvement heuristic, we suggest improving the neighborhood operator. The current neighborhood operator is not able to decrease the overtime close to the amount of overtime used in the MIP solution. Due to time limitations and the fact that the MIP model can be used for the standard planning horizon of HA, this is not assessed during this research.

Implementing the heuristics in a more suitable computer program. During this research, the heuristics are implemented in AIMMS. AIMMS software is especially for designing and solving mathematical models. However, it is not designed to solve heuristics. More appropriate programs exist for coding and solving heuristics such as Delphi or Python. Due to time limitations and the focus on the MIP model, we were not able to implement the heuristics into Delphi or Python.

Improving data of HA. The models heavily rely on the data of HA in the ERP system. By improving the data, the reliability of the MIP model improves. The chance of achieving the model solutions, in reality, is higher when accurate data is used. Most important is to update the product routings and their processing times.

Investigating the impact of a rolling planning horizon on the models. We assume no overlap between 2 consecutive planning horizons, whereas in practice this may happen. Due to time limitations, the impact of a rolling planning horizon is not researched. We recommend researching the impact in a later stage.

Limit the impact of the bottleneck operations. The research shows four bottleneck operations: SMD, SSOL, AOI and PR. We recommend researching how to limit the bottleneck operations such that there are no bottleneck operations in the future. One possibility is to research if regular capacity needs to be extended to cope with the bottlenecks in the future.

Investigating the use of the MIP model at order acceptance. Resource loading can also be used at the order acceptance phase to gain insights into the resource capacity levels and to gain insights into the impact of accepting an incoming order. By using resource loading earlier in the process, high costs for overtime and tardiness can be limited since resource capacity levels are already checked at the order acceptance phase and preventive measures can be taken such as rejecting the order. We recommend investigating the use of the MIP model in order acceptance to limit the cost of overtime and tardiness.

6.3 Contribution to theory and practice

The main contribution of this research to theory is the resource loading MIP model combining the resource-driven and time-driven approaches. Literature on resource loading is limited especially for combining the two approaches. The resource loading model proposed in Hans (2001) corresponds most to our proposed MIP model. A major distinction between the model of Hans (2001) and our proposed model is that Hans (2001) allows preemption, whereas our proposed model does not allow preemption within the operations of an order. To the best of our knowledge, there is no resource loading MIP model available in literature that combines the resource-driven and time-driven approach taking into account non-preemption constraints, capacity constraints and precedence constraints simultaneously.

Another contribution to theory is the description of the job shop production environment. In most traditional job shop planning and scheduling problems, the problem consists of jobs J_j that are planned on machines M_m in such a way that the capacity of the machines is not exceeded. The job shop production environment in this research has orders I_i with a unique product routing containing some of the operations J_j, where S_{is} indicates the s-th operation in the product routing of order I_i. Each operation J_j has its capacity. Each S_{is} is planned in such a way that the capacity of the corresponding operation J_j is not exceeded and the precedence constraint of an order is not violated. The job shop production environment of this research can be used by other researchers modelling a similar production environment. The proposed MIP model can be used by researchers modelling a resource loading problem based on the resource-driven and time-driven approach with non-preemption, capacity and precedence constraints for a similar job shop production environment. Also, specific parts of the model can be used as an inspiration for other researchers such as modelling of the non-preemption constraints.

This research contributes to practical applications by introducing a resource loading method that integrates resource capacity into HA's tactical production planning. HA did not differentiate between strategic, tactical, and operational planning. Following this research, HA has acquired the capability for effective tactical planning. For planning horizons of up to 2 months, HA can employ the proposed MIP model to generate an optimal production plan, yielding insights into tardy orders and required overtime hours in a cost-efficient manner. Another contribution is demonstrating HA's potential to improve its Customer Service Level beyond 90% at the expense of overtime costs. Furthermore, the research demonstrates the bottleneck operations SMD, SSOL, AOI and PR. HA can further investigate how to deal with those bottleneck operations such that overall operational efficiency and capacity utilisation can be achieved.

6.4 Implementation challenges

Implementing the proposed model at HA includes some challenges. This section explains possible implementation challenges.

Integration with existing systems. One of the main challenges involves integrating the proposed model into existing systems. The current MIP model is developed in AIMMS, a high-cost software provided by the University of Twente. To adopt the solution, HA can acquire an AIMMS license. Obtaining an AIMMS license allows for modest model adjustments but requires a business license investment. Alternatively, HA could use free software like Anaconda and design the model using Python. This option requires specific Python knowledge. For both options, the main concern is the integration of the software with Isah which contains essential data to perform the model. Integration with Isah is possible via SQL but it needs to be designed such that the planner can quickly and easily execute the model for the scenarios he needs. A more optimal but expensive option is integrating the MIP model directly into Isah. This option allows the planner to use one software instead of Isah combined with AIMMS or free software. HA works with an ERP consultant who is able to implement customised programs into Isah at the expense of time and cost. A recommendation for HA is to discuss the possibilities with the ERP consultant and make a trade-off on the option in terms of the user-friendliness of the solution, implementation time and implementation cost.

Resistance of employees for the proposed solution. Another challenge is the resistance of employees. The operations manager has employed production planning for almost 20 years and might be used to his method and may be resistant to change. A recommendation for HA is to use clear communication about the benefits of the model, addressing the concerns of the stakeholders and involving the stakeholders in the implementation process.

References

E.H.L. Aarts, J.K. Lenstra, P.J.M. Laarhoven van, and N.L.J. Ulder. *Job shop scheduling by local search*. Memorandum COSOR. Technische Universiteit Eindhoven, 1992.

Mageswaran Arunasalam, Zulkiflle Leman, B.T.H.T.B. Baharudin, Shamsuddin B. Sulaiman, and Charles Santhakumar Anthony Das. Challenges in minimizing copper dissolution for lead free wave soldering in surface mount technology going towards green manufacturing. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 9:645–660, 03 2022. doi: 10.1007/s40684-020-00298-0.

Kaouthar Cherkaoui, Robert Pellerin, Pierre Baptiste, and Alain Haït. A time driven rccp model with two levels of planning and a reactive planning approach for tactical project planning. *Procedia Computer Science*, 64:257–264, 12 2015. doi: 10.1016/j.procs.2015.08.488.

Noud Gademann and Marco Schutten. Linear-programming-based heuristics for project capacity planning. *IIE Transactions*, 37(2):153–165, 2005. doi: 10.1080/07408170590885611. URL `https://doi.org/10.1080/07408170590885611`.

Eduardo Guzman, Beatriz Andres, and Raul Poler. Models and algorithms for production planning, scheduling and sequencing problems: A holistic framework and a systematic review. *Journal of Industrial Information Integration*, 27:100287, 2022. ISSN 2452-414X. doi: https://doi.org/10.1016/j.jii.2021.100287. URL `https://www.sciencedirect.com/science/article/pii/S2452414X21000844`.

Erwin Hans. *Resource Loading by Branch-and-Price Techniques*. PhD thesis, 01 2001.

Hans Heerkens and Arnold van Winden. *Solving Managerial Problems Systematically*. Noordhoff Uitgevers, 2017. ISBN 9789001887957. Translated into English by Jan-Willem Tjooitink.

Oliver Holthaus and Chandrasekharan Rajendran. Efficient dispatching rules for scheduling in a job shop. *International Journal of Production Economics*, 48(1):87–105, 1997. ISSN 0925-5273. doi: https://doi.org/10.1016/S0925-5273(96)00068-0. URL `https://www.sciencedirect.com/science/article/pii/S0925527396000680`.

Byoungil Jeon, Jongyul Kim, Kiseo Lim, Younghyun Choi, and Myungkook Moon. Calculation of detector positions for a source localizing radiation portal monitor system using a modified iterative genetic algorithm. *Journal of Radiation Protection and Research*, 42:212–221, 12 2017. doi: 10.14407/jrpr.2017.42.4.212.

kolb Cleaning Technology GmbH. Pcb cleaning (cleaning of assemblies). Available at `https://www.kolb-ct.com/cleaning-tasks/product-cleaning/pcb-assemblies-cleaning` (2022/08/24), n.d.

Eduardo Lalla. Branch & bound. Course Operations Research Techniques 1 at the University of Twente, March 2021.

Ning-Cheng Lee. 1 - introduction to surface mount technology. In Ning-Cheng Lee, editor, *Reflow Soldering Processes*, pages 1–18. Newnes, Burlington, 2001. ISBN 978-0-7506-7218-4. doi: https://

doi.org/10.1016/B978-075067218-4/50001-4. URL https://www.sciencedirect.com/science/article/pii/B9780750672184500014.

Gréanne Maan-Leeftink. Constructive heuristics. Course Operations Research Techniques 1 at the University of Twente, March 2021a.

Gréanne Maan-Leeftink. Improvemnt heuristics part i. Course Operations Research Techniques 1 at the University of Twente, March 2021b.

Gréanne Maan-Leeftink. Improvemnt heuristics part ii. Course Operations Research Techniques 1 at the University of Twente, March 2021c.

MPE-electronics. The essential guide to coatings and potting in pcb assembly: Conformal coating. Available at https://www.mpe-electronics.co.uk/2021/02/09/the-essential-guide-to-coatings-and-potting-in-pcb-assembly-conformal-coating (2022/08/24), a.

MPE-electronics. The essential guide to coatings and potting in pcb assembly: Potting. Available at https://www.mpe-electronics.co.uk/2021/04/13/the-essential-guide-to-coatings-and-potting-in-pcb-assembly-potting (2022/08/24), b.

Andrés Muñoz-Villamizar, Javier Santos, Jairo Montoya-Torres, and M.J. Alvarez. Improving effectiveness of parallel machine scheduling with earliness and tardiness costs: A case study. *International Journal of Industrial Engineering Computations*, 02 2019. doi: 10.5267/j.ijiec.2019.2.001.

Kay Peeters and Henny van Ooijen. Hybrid make-to-stock and make-to-order systems: a taxonomic review. *International Journal of Production Research*, 58(15):4659–4688, 2020. doi: 10.1080/00207543.2020.1778204. URL https://doi.org/10.1080/00207543.2020.1778204.

Michael L. Pinedo. *Scheduling: Theory, Algorithms, And Systems*, chapter 14. Springer Cham, Switzerland, 5 edition, 2016a. ISBN 978-3-319-26580-3.

Michael L. Pinedo. *Scheduling: Theory, Algorithms, And Systems*, chapter 2. Springer Cham, Switzerland, 5 edition, 2016b. ISBN 978-3-319-26580-3.

David J. Rader. Deterministic operations research: Models and methods in linear optimization. 2010.

Kelli R. Robinson and Scott A. Moses. Effect of granularity of resource availability on the accuracy of due date assignment. *International Journal of Production Research*, 44(24):5391–5414, 2006. doi: 10.1080/00207540600665810. URL https://doi.org/10.1080/00207540600665810.

Tadeusz Sawik. Hierarchical approach to production scheduling in make-to-order assembly. *International Journal of Production Research*, 44(4):801–830, 2006. doi: 10.1080/00207540500340969. URL https://doi.org/10.1080/00207540500340969.

Anouk Scholten. Optimizing the inventory control policy within an electronics production company, 2020. URL https://essay.utwente.nl/82652/1/Master_thesis__Final%20Report__Anouk_Scholten.pdf.

J.M.J. Schutten. *Shop floor scheduling with setup times.* PhD thesis, University of Twente, 1996.

Leonardo Tavares, Heitor Lopes, and Carlos Erig Lima. Construction and improvement heuristics applied to the capacitated vehicle routing problem. pages 690–695, 01 2009. doi: 10.1109/NABIC.2009.5393467.

Matthias Thurer, Mark Stevenson, Cristovão Silva, Martin Land, Lawrence Fredendall, and Steven Melnyk. Lean control for make-to-order companies: Integrating customer enquiry management and order release. *Production and Operations Management*, 23, 03 2014. doi: 10.1111/poms.12058.

Matthias Thürer, Mark Stevenson, Cristovao Silva, and Martin Land. Towards an integrated workload control (wlc) concept: The performance of due date setting rules in job shops with contingent orders. *International Journal of Production Research*, 51(15):4502–4516, 2013. doi: 10.1080/00207543.2013.774485. URL `https://doi.org/10.1080/00207543.2013.774485`.

Paul Vianco and Y Feng. *Electronic Packaging: Solder Mounting Technologies.* 01 2016. ISBN 9780128035818. doi: 10.1016/B978-0-12-803581-8.10099-2.

Chung-Hsing Yeh. A customer-focused planning approach to make-to-order production. *Industrial Management and Data Systems*, 100:180–187, 06 2000. doi: 10.1108/02635570010328693.

Appendices

A Solution design: technical details constructive algorithm

We introduce the following auxiliary parameters to perform step 2.2 in Figure 4.3: $TotalUsedOverTime_{jt}$, $TotalUsedRegular_{jt}$, $AvailableRegular_{jt}$, $AvailableOvertime_{jt}$, $StillToSchedule_{ist}$, $ScheduleRegular_{ist}$ and $ScheduleOvertime_{ist}$. The following points are evaluated for each time bucket t in the ATW window until S_{is} is completely planned or the complete ATW window is evaluated:

1. **The amount of regular capacity left in time bucket $t \rightarrow$ 3 possible scenarios:**
   (a) $AvailableRegular_{jt} = 0$
   (b) $AvailableRegular_{jt} < StillToSchedule_{ist} \rightarrow$ update parameters:
      i. $ScheduleRegular_{ist} = AvailableRegular_{jt}$
      ii. $StillToSchedule_{ist} = StillToSchedule_{ist} - ScheduleRegular_{ist}$
      iii. $TotalUsedRegular_{jt} = TotalUsedRegular_{jt} + ScheduleRegular_{ist}$
      iv. $AvailableRegular_{jt} = 0$
   (c) $AvailableRegular_{jt} \geq StillToSchedule_{ist} \rightarrow$ update parameters:
      i. $ScheduleRegular_{ist} = StillToSchedule_{ist}$
      ii. $StillToSchedule_{ist} = 0$
      iii. $TotalUsedRegular_{jt} = TotalUsedRegular_{jt} + ScheduleRegular_{ist}$
      iv. $AvailableRegular_{jt} = c_j - TotalUsedRegular_{jt}$
2. **The amount of overtime left in time bucket $t \rightarrow$ 5 possible scenarios:**
   (a) No overtime left and $ScheduleRegular_{ist} = 0$
   (b) No overtime left but $ScheduleRegular_{ist} > 0$
   (c) $AvailableOvertime_{jt} < StillToSchedule_{ist}$ and $ScheduleRegular_{ist} = 0$
   (d) $AvailableOvertime_{jt} < StillToSchedule_{ist}$ but $ScheduleRegular_{ist} > 0$
   (e) $AvailableOvertime_{jt} \geq StillToSchedule_{ist} \rightarrow$ update parameters:
      i. $ScheduleOvertime_{ist} = StillToSchedule_{ist}$
      ii. $StillToSchedule_{ist} = 0$
      iii. $TotalUsedOverTime_{jt} = TotalUsedOverTime_{jt} + ScheduleOvertime_{ist}$
      iv. $AvailableOvertime_{jt} = mo_{jt} - TotalUsedOverTime_{jt}$
3. **The number of hours already planned for all $s \in S_i$ in time bucket t:**
   $PercentageWeek_{it}$ provides the percentage of the time bucket already used by order i, i.e. $PercentageWeek_{it} = PercentageWeek_{it} + (ScheduleRegular_{jt} + ScheduleOvertime_{jt}) / (c_j + mo_{jt})$. If $PercentageWeek_{jt} > 1$, the parameters are set back to their previous value and the next $t := t - 1$ is evaluated. For example, $S_{12} = J_4$ with $p_{12} = 10$. S_{12} is completely planned at $t = 5$ with $ScheduleRegular_{45} = 8$ and $ScheduleOvertime_{45} = 2$. The maximum regular capacity and overtime are $c_4 = 36$ and $mo_{45} = 9$. S_{12} is the first operation of order i planned in $t = 5$. $PercentageWeek$ is calculated by $(8 + 2)/(36 + 9) \approx 0.22$. $1 - 0.22 \approx 0.78$ is left in time bucket $t = 5$ to plan other operations of order i.