UNIVERSITY OF TWENTE.

Bachelor Thesis

# Improving the Service Process using Simulated Annealing at Company X


L.M. Bettonvil s2154005

**First Supervisor University**
dr. H. Chen (Hao)

**Second Supervisor University**
dr. L. Xie (Lin)

**Company Supervisor**
ir. T.J. Bemthuis (Thijs)


**Date**
April 30, 2024

## Preface

Dear reader,

I am delighted to present this bachelor thesis, titled *Improving the Service Process using Simulated Annealing at Company X*, which represents the culmination of my academic work at the University of Twente. This work delves into improving the maintenance department at Company X by employing job scheduling and heuristics.

First and foremost, I would like to extend my heartfelt gratitude to Thijs Bemthuis from Company X for his support and guidance throughout this project. His expertise and dedication have been invaluable in shaping the direction and scope of my research. His insights and encouragement have enriched my learning experience and inspired me to strive for excellence in my academic pursuits.

I want to thank dr. Hao Chen, my first university supervisor, for his scholarly guidance, mentorship, and support. Dr. Chen's knowledge, constructive feedback, and encouragement have played a pivotal role in refining my research methodology and enhancing the quality of my analysis. I also want to thank dr. Lin Xie, my second university supervisor, for the constructive feedback and guidance. I am genuinely grateful for their support in completing my bachelor of Industrial Engineering and Management at the University of Twente.

Additionally, I would like to sincerely thank my friends and family for their unwavering support, encouragement, and understanding throughout this academic endeavour. Their constant support and belief in my abilities have been a source of strength and motivation during this journey's challenges and triumphs.

Enjoy reading my bachelor thesis!

Thank you.

Lars Bettonvil

April 30, 2024

## Abstract

This research concerns Company X, a landscaping equipment retailer and servicer based in Location Y. It aims to develop a service scheduling method through an in-depth analysis. The method is developed by contextualising the current situation and studying relevant theories. These theories are used to create a schedule optimization model during the solution design phase. The model is then tested and evaluated to provide Company X with recommendations and further research suggestions.

The literature review revealed that a Simulated Annealing (SA) metaheuristic approach was effective in similar cases to Company X. This SA metaheuristic is combined with the current scheduling (FCFS) approach and the Shortest Processing Time (SPT) and Longest Processing Time (LPT) heuristics. The best input settings for the SA metaheuristic were chosen during the evaluation by conducting an experiment for each setting.

Comparing FCFS, SPT, and LPT showed that the SPT performed 2.72% worse in total makespan, and the LPT performed 1.20% better compared to the FCFS heuristic. Comparing each of the constructive heuristics combined with the SA metaheuristic to the current FCFS heuristics showed that the FCFS + SA, SPT + SA, and LPT + SA heuristics performed 8.33%, 5.39%, and 5.89% better with regard to the total makespan, respectively.

This research is significant in exploring the possibilities of heuristic improvement in the service industry. Although heuristics are well-researched in other sectors, little research has been conducted on their use in the servicing sector. Therefore, this research explores the possibilities of heuristic improvement.

## Management Summary

This research is conducted at Company X, a professional landscaping equipment retailer and servicer in Location Y. They position themselves open to a wide range of clientele, from private individuals to municipalities and utility companies. Company X relies heavily on its aim to provide excellent service and prolong the life of its products to achieve a sustainable future. For this research, the aim was set to analyse and improve the service process of the maintenance department and make adjustments accordingly. During the initial investigation, it was found that the maintenance department could not anticipate equipment function losses, which resulted in sub-optimal utilisation of time resources in the workshop. This, in turn, meant that the overall servicing process was inefficient, and the mechanic capacity was not optimally utilised. This resulted in the main research question:

*"What is the most effective approach to developing an improved service schedule method that Company X can use to improve the maintenance process and reduce service times?"*

The servicing process should be analysed first to develop a service scheduling method. The overall servicing process was analysed from when the customer contacted customer service at Company X to when the customer got their equipment back and continued working with the repaired equipment. A BPMN was modelled to get an overview of the internal tasks for a single service order, followed by a detailed analysis of historical service order data. Each service order consists of two aspects: a regular service job and extra service time, which isn't considered as one of the regular service jobs. The data regarding the regular service jobs was gathered through data analysis and mechanic interviews. First, these regular jobs were analysed based on occurrences, after which mechanics interviews were held to understand the servicing process better; during these interviews, the regular service job times were gathered and analysed. Next, the extra service time was analysed. This additional time is manually registered on a service order by the mechanic, and this is also the time a customer pays for the performed maintenance (if it does fall not one of the standard jobs). Due to the wide variety of equipment serviced by the maintenance department, categories were determined in which each service order would fit. Based on these categories, several descriptive statistics were analysed. The averages of these service times were chosen as the input dataset for the formulated scheduling method.

Next, a literature study was conducted in which a theoretical framework was drafted to give a theoretical perspective to the problem at hand. First, both job scheduling problems and the knapsack problems were introduced. The scheduling problems were analysed, and different cases of scheduling problems could apply to the case of Company X. Then, knapsack problems are introduced; these knapsack problems are related to the physical analogy of fitting items with a certain weight into a knapsack until a specific weight capacity is reached. The comparison with a workshop mechanic is made as the capacity of the knapsack can be seen as the working hours of the mechanic, and the weights of the items can be seen as the service times of the service orders. Different variants were mentioned for the knapsack problem as well. Next, the attention was turned towards how these problems could be solved. The first optimal way of solving these kinds of problems is through exact algorithms; different exact algorithms were proposed, but all exact algorithms have the same negative aspect. They are computationally intensive and cannot be used for significant problems. Therefore, another approach was discussed: the use of heuristics. Heuristics are a set of rules on which decisions should be made regarding developing a service schedule. Metaheuristics were also introduced, which are used to improve the initial schedule. Analysing case studies, the Simulated Annealing metaheuristic for scheduling problems was found to be similar to the maintenance department. Therefore, the SA metaheuristic was chosen as the schedule improvement method.

After this, the solution model was designed and discussed. First, a problem description was given, including parameters, completion time, objective function and constraints. Then, the requirements and

assumptions for the solution model were discussed, leading to the use of heuristics. Three constructive heuristics were chosen to formulate an initial solution in the solution model; these heuristics are the current scheduling method (FCFS), Shortest Processing Time (SPT) and Longest Processing Time (LPT) heuristics. The Simulated Annealing (SA) metaheuristic is also discussed before moving to the functions of the scheduling model. In the description of the scheduling model, all necessary microflows of the Mendix application were discussed, and the inner workings of the Mendix application were shown. During this stage, the Mendix application is also tested to determine if the output of the application is correct.

The scheduling model was evaluated by analyzing experimental design, experiments, and heuristic evaluations. The experiments focused on the input settings of the Simulated Annealing metaheuristic and aimed to determine the optimal values for $T_{start}$, $T_{stop}$, $\alpha$, $k$ and $n$. The completion and computation time was used to evaluate the input settings. A proof of concept was presented to show that the SA metaheuristic performed as intended. The heuristics were assessed using the selected input settings for the SA metaheuristic.

First, the three constructive heuristics were evaluated. It was found that SPT heuristics performed 1.14% and 2.72% worse than the FCFS approach on the daily average idle time per mechanic and total makespan KPIs, respectively. The LPT heuristic performed 10.47% and 1.20% better than the FCFS approach over the same respective KPIs.

After this, the three constructive approaches were combined with the SA metaheuristic, each formulating an initial solution. The SA metaheuristic was then run with the input settings from the experiments. These three metaheuristics were compared to the current (FCFS) scheduling approach. It was found that the FCFS + SA, SPT + SA, and LPT + SA metaheuristics performed 8.33%, 5.39%, and 5.89% better than the FCFS scheduling approach. Compared to the LPT heuristic, they performed 7.22%, 4.24%, and 4.75% better on the completion time KPI, respectively.

Furthermore, the heuristics were evaluated weekly, and an implementation plan was proposed to incorporate the Mendix application into daily operations.

Lastly, conclusions and recommendations were made based on the results in the preceding chapters. It was found that the scheduling model had quite a lot of emptiness, causing a lot of idle time. First, this could be caused by incorrect hour registration, as the service times come directly from the hours of the service invoices. Based on the results, several recommendations were formulated:

- Implement a dedicated scheduling policy
- Develop an integrated schedule application
- Investigate service time registration

This research has successfully developed and applied a schedule improvement method to the maintenance department. The study gathered contextual and theoretical insights on the servicing process and utilised the Simulated Annealing metaheuristic to improve the service schedule. Overall, the research demonstrates the effectiveness of the proposed method in enhancing the maintenance department's scheduling process.

# Contents

# 1 Introduction

Job scheduling has become increasingly popular in recent years [1] due to its potential to improve organisational efficiency. Efficiency in this context refers to the efficient use of resources, and this research explicitly refers to the use of time resources in the maintenance department. By streamlining operations, job scheduling can also lead to additional benefits, such as gaining a competitive edge in customer relations and increasing overall profitability.

Heuristics offer a significant advantage over exact algorithms in terms of computational efficiency. While exact algorithms can be computationally intensive, heuristics can solve scheduling problems using reasonable computational resources [2]. This section introduces the research and the company where it was conducted. It further delves into the research design and methodologies on which the rest of the thesis is based, primarily focusing on improving the servicing process.

## 1.1 Company Description

The research centres around Company X, a company in Location Y that specialises in retailing and servicing high-quality gardening tools and machinery. Company X caters to a diverse clientele, ranging from private individuals to large organisations such as municipalities and utility companies, all united by their demand for top-tier products. One of the critical factors influencing customers to choose Company X is its emphasis on servicing, a crucial aspect given the often heavy-duty nature of the equipment they provide.

Aligned with its future vision, Company X is committed to forging a sustainable future for its customers. The efficient maintenance offered to their clientele is pivotal in realising this vision. By ensuring the prolonged lifespan of their products and mitigating the need for new equipment, Company X recognises significant financial benefits and contributes to a reduced environmental impact. This approach to equipment maintenance gains significance amid a growing societal emphasis on community-wide waste reduction, driving increased demand for product longevity and, consequently, enhanced maintenance services.

Company X recognises this evolving need in their daily operations, observing a rising trend where the equipment serviced by the company appears prematurely or delayed. Recognising the importance of gaining insights into the servicing process and identifying areas for improvement, Company X aims to minimise waste, ultimately making a substantial impact on product lifespan and environmental sustainability.

## 1.2 Problem Identification

This section will discuss the problem identification of the proposed bachelor research. Firstly, it will examine the action problem followed by the problem cluster and core problem. Lastly, it discusses the norm and reality of the core problem, the current situation, and the optimal situation the research aims to reach.

### 1.2.1 Action Problem

Company X is currently grappling with a notable challenge in the timing of client equipment servicing. Frequently, the servicing department encounters equipment that undergoes maintenance either prematurely—where no parts necessitate replacement—or too late when the equipment is incapacitated and in need of urgent repair. The former scenario leads to inefficiencies in the servicing department, like waiting times for external vendors and storage capacity for equipment, reflecting shortcomings in lean management practices. Conversely, the latter results in prolonged downtime for clients' equipment as they await inspection and repairs within the servicing department. This problem leads to equipment function loss on the client's side, meaning that Company X' clients cannot perform their jobs as they experience equipment malfunctions.

Ideally, Company X envisions a scenario where client equipment management aligns precisely with the optimal timing for maintenance, resulting in Company X' clients never experiencing any equipment function loss. In this ideal scenario, the servicing department's planning could seamlessly schedule maintenance for all equipment, eliminating the current challenges experienced in planning at Company X' servicing department. This alignment would enhance operational efficiency and provide a more streamlined and effective maintenance process. Addressing this action problem is pivotal for Company X to improve its service delivery and client satisfaction while improving resource utilisation within the servicing department. For those reasons, as mentioned earlier, the following action problem is formulated:

*"Company X maintenance department cannot anticipate equipment function losses, which need maintenance"*

### 1.2.2 Problem Cluster & Core Problem

As the action problem is now defined, we can look further towards the core problem of this situation at Company X to formulate the correct core problem for this research. The action problem is coherent with another big issue. Company X is currently struggling with downtime for client equipment, meaning that the equipment cannot be used before any job. This, in turn, is caused by inefficiencies regarding the servicing schedule, which are caused by a lack of spare part inventory and the resulting lead times for those spare parts. Furthermore, these schedule inefficiencies are caused by unnecessary maintenance, resulting in decreased employee utilisation, as the work performed doesn't contribute to anything. These inefficiencies are formulated as unnecessary or sub-optimal utilisation of resources, for the case of Company X working time for the mechanics.

The problem cluster, Figure 1, shows that three main problems cause the core problem. First of all, both instances where clients' equipment is being serviced either prematurely or too late. Also, the problem is that Company X' servicing department does not know the status of the client's equipment. These three problems are the leading cause of the core problem at Company X' servicing department core problem:

*"Inefficient servicing process and sub-optimal utilisation of mechanic capacity, resulting in an inefficient service schedule"*

### 1.2.3 Gap Between Norm & Reality

Company X' current reality is a sub-optimal utilisation of time resources resulting in an inefficient scheduling policy in which they manage their servicing process: a product comes in for servicing, gets inspected, if necessary, spare parts are ordered, and service is done. They result in many problems discussed in Section 1.2.2. The gap between the current reality and the desired situation is that they don't have a way to deal effectively with the inefficient scheduling policy. This research aims to provide a way to change the service schedule to schedule incoming service orders more efficiently. Therefore, this research will improve the servicing process at Company X, working towards the norm of efficient servicing.

## 1.3 Research Design

This section will introduce the main research question, including the motivation for the question and division into multiple sub-questions. Afterwards, the research's problem-solving approach, de-

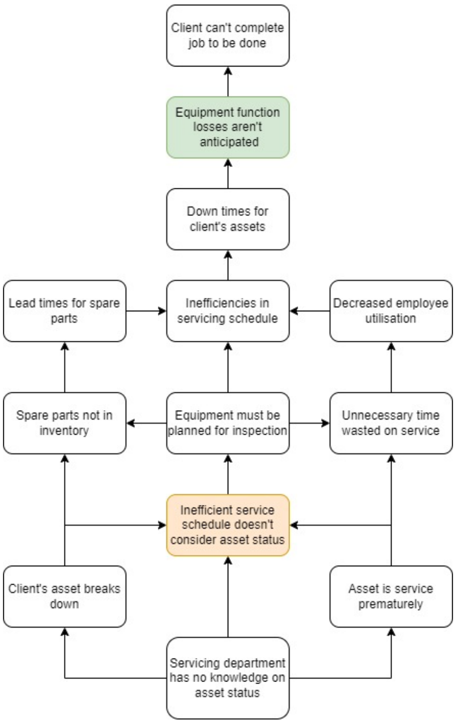


Figure 1: Problem cluster

liverables, and intended scope will be discussed. This will be followed by the research's limitations, validity & reliability.

### 1.3.1 Problem-Solving Approach

The research will use the Managerial Problem-Solving Method (MPSM) introduced by Heerkens & van Winden [3]. This method combines creative and systematic approaches to solve complex problems, particularly those in the organisational context [3]. Since the core problem of this research involves changing the servicing process and schedule, which is an organisational problem, the MPSM is an appropriate methodology. It is designed to solve practical problems that require creative formulation of solutions. As discussed in Section 1.2.2, a servicing strategy's selection and implementation process is a reasonably complex problem. Hence, the MPSM approach is the right choice for this bachelor's research. The MPSM consists of seven steps combining systematic and creative approaches to achieve the best possible research process.

1. Defining the problem
2. Formulating the approach
3. Analysing the problem
4. Formulating (alternative) solutions
5. Choosing a solution
6. Implementing the solution
7. Evaluating the solution

## 1.4 Research Questions

Formulating a practical research question is pivotal to achieving a structured approach for this bachelor's research. In this section, the main research question, which will provide an answer to the core problem, is developed. This primary research question will be split into sub-questions to shed light on all aspects of the proposed bachelor research. The main research question is derived from the problem cluster in Section 1.2.2

*What is the most effective approach to developing an improved service schedule method that Company X can use to improve the maintenance process and reduce service times?*

### 1.4.1 Sub-Questions

This section will formulate the sub-questions used for the problem-solving approach with an explanation and motivation for each sub-question (SQ). Each of the sub-questions is linked with the research stages of the MPSM [3]; these stages are defined as: *Analysing the problem*, *Formulating & Choosing a solution*, *Implementing the solution* and *Evaluating the solution*

**Stage 1: *Analysing the Problem***

This stage will analyse the core problem and the current situation at Company X' maintenance department. A description of how Company X handles their servicing schedule must be sought. Utilising data from the Company X ERP system, this process will be analysed. Next, interviews with Company X employees will be held to gain insight into the servicing process, from scheduling to maintenance itself. The information gathered from the interviews will be formulated into a process model that summarises and analyses business process aspects. Here, the service times from different product categories will be analysed and criteria for a scheduling method will be formulated. Gathering this information for the context analysis results in the first sub-question

SQ1: *How is Company X currently handling their service operations, and what are the service times associated with these operations?*

**Stage 2: *Formulating & Choosing a Solution***

Secondly, exploration is needed to learn how servicing operations can be managed. This will employ an academic literature search and review to see how the scheduling method can be designed in Company X' maintenance department. Furthermore, a scheduling method that best fits Company X' situation will be chosen, which results in the next sub-question.

SQ2: *What relevant theories best apply to the maintenance department to develop a schedule improvement method?*

**Stage 3: *Implementing the Solution***

The third stage will implement the chosen scheduling method into Company X' situation. Here, a description of how the scheduling method can be implemented into Company X' servicing department operations will be given. The chosen solution method will be implemented into a Mendix model. This methodology was selected as Company X uses Mendix in their daily operations, giving it excellent integrability as a possible scheduling method.

SQ3: *How can Company X develop a service schedule using the selected scheduling method?*

**Stage 4: *Evaluating the Solution***

The developed scheduling model and its complementary implementation method must be evaluated to see if the research works and achieves the intended research goal. After this, conclusions can be formulated, and further recommendations can be made. The resulting sub-question therefore is

SQ4: *What recommendations and conclusions can be made after implementing the servicing methodology and schedule?*

## 1.5 Deliverables

After discussion, the conclusion was drawn that the following intended deliverables have to be set for this bachelor's research:

- Contextual analysis of the current situation at Company X
- Description of schedule methodology
- Mendix scheduling model
- Advisory with an implementation strategy for the scheduling model
- Further recommendations

## 1.6 Research Goal

This research aims to gain knowledge to improve the servicing process. By developing a scheduling method, this research should develop a way to reduce service times. The knowledge gained from this study will be applied to the case study of Company X, hopefully leading to a reduction in service times at Company X' servicing department. Ultimately, this reduces the downtime for clients' equipment, resulting in satisfied customers.

### 1.6.1 Scope

To ensure proper research, a scope must be set; this will ensure strict boundaries for the research, thereby preventing unnecessary time wasted on irrelevant topics. This research aims to help Company X' servicing department improve their servicing processes (i.e., reducing service times). Therefore, within this research's scope is the evaluation of the current process, the derivation of a schedule improvement method and the formulation of an implementation strategy for said scheduling model. Due to time constraints, not within this research's scope are long-term consequences of the proposed research.

## 1.7 Validity & Reliability

Validity and reliability are two essential factors in research; some may become utterly irrelevant without considering these factors. Saunders et al. define validity as the "Extent to which data collection methods or methods accurately measure what they are intended to measure."[4] Meaning that a researcher should consider if the appropriate data collection methods are used and especially consider the accuracy of analysis of the results. Considering this will ensure that the results of this research will ensure that a real-world application is feasible.

In turn, reliability is defined as the "extent to which data collection technique or techniques will yield consistent findings, similar observations would be made or conclusions reached by other researchers or there is transparency in how sense was made from the raw data." [4] So reliability aims to ensure that the research produces consistent results, even when the same research is repeated. It also considers that assumptions or biases from the researcher may cause unreliability.

For this bachelor research, these aspects are considered in the fourth stage of the proposed research design. Section 5 shows that this stage will evaluate if the research has fulfilled its intended goal and consider whether the results are valid and reliable. However, it is crucial to consider these factors closely; otherwise, the research may result in incorrect conclusions and recommendations towards Company X' servicing department.

# 2 Context Analysis

This chapter will discuss the analysis centred around the contextual performance of the current situation. It is performed to gain information on the servicing process at Company X' servicing department. The entire process is discussed in Section 2.1, followed by the current scheduling method in Section 2.2. Next, Section 2.3 discusses the data analysis performed. After which, Section 2.4 considers the scheduling characteristics. This is followed by a discussion of assumptions, constraints and Key Performance Indicators (KPIs) in Sections 2.5, 2.6 and 2.7, respectively. Section 2.9 summarises this chapter to answer its sub-question:

*How is Company X currently handling their service operations, and what are the service times associated with these operations?*

## 2.1 Process Analysis

As mentioned in Section 1.2.3, the servicing process consists of a basic process. All aspects of the maintenance process will be included in the SIPOC models. SIPOC models are used to model the process on a high level, giving a clear overview. A SIPOC model utilises a process's Supplier, Input, Process, Output and Customer to provide a clear general view of what occurs during a process [5]. This is especially useful when starting to solve a research problem. The process as a whole can be depicted into five separate sub-processes (see Figure 2), being *Function loss observation*, *Inbound logistics*, *Maintenance process*, *Outbound logistics* and *Resolved function loss*.

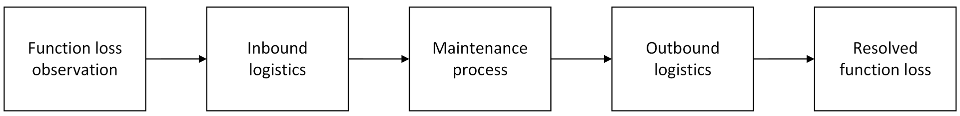


Figure 2: Process map

This general process starts with a *function loss observation*, which all centres around the equipment and the end user. A function loss occurs when the end user of a piece of equipment cannot perform their scheduled tasks with the equipment. This results in delays at the end user's end. When an equipment failure occurs, the equipment's user notices this occurrence, which leads to the observation of the equipment function loss, as shown in Figure 7.

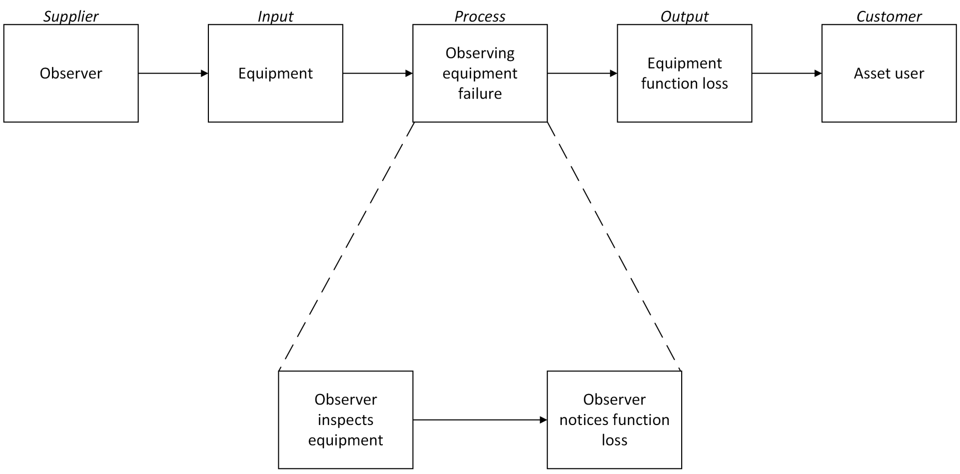


Figure 3: Equipment Failure Process

The *Inbound Logistics* process follows immediately once a user experiences a functional loss and requires a solution. To resolve the issue, the user must contact Company X' customer service to suggest possible solutions. The customer service agent will then inform the customer about all the available servicing options. The customer is responsible for deciding on the required service and creating a service order for their equipment through customer service. If the customer uses Company X' internal logistics system, their equipment can be scheduled for transportation; otherwise, the customer must bring it to the maintenance department. It is crucial to note that the equipment cannot be transported to Company X' maintenance department in extraordinary cases. In such cases, an appointment will be made for on-site maintenance. Following the *Inbound Logistics* process is essential to ensure efficient and prompt service delivery. The process is illustrated in Figure 4.

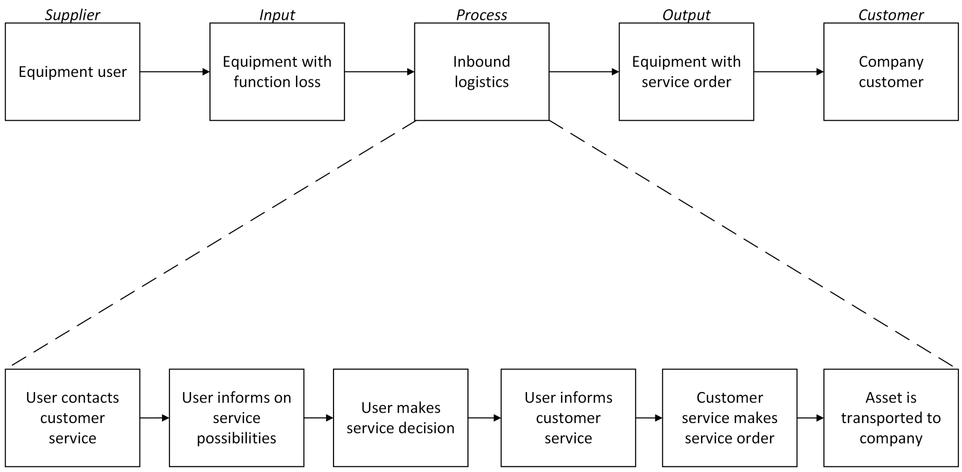


Figure 4: Inbound logistics process

Next is the *Maintenance process*; this process starts with a customer needing to service a piece of equipment. The equipment is brought to the maintenance department, specifically to a mechanic, who inspects the equipment and decides the course of action to resolve the experienced function loss. If necessary, spare parts, tools, or other materials are picked, and the equipment is repaired. Lastly, the equipment is put back together and ready to return to the customer.

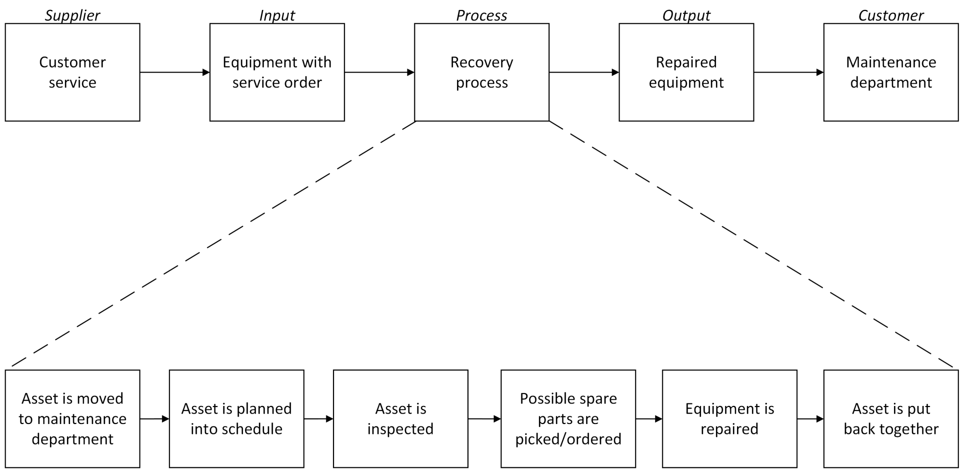


Figure 5: Maintenance process

Specific final tasks must be completed before returning the equipment to the customer to complete the Outbound Logistics process (as shown in Figure 6). Firstly, the service order needs to be closed, and an e-mail notification should be automatically sent to the customer informing them that their equipment is now available for pick-up. If required, the customer can request internal logistics service and a delivery date will be arranged accordingly. Finally, the customer service team will issue an invoice for the completed repairs, which will be directly sent to the customer.

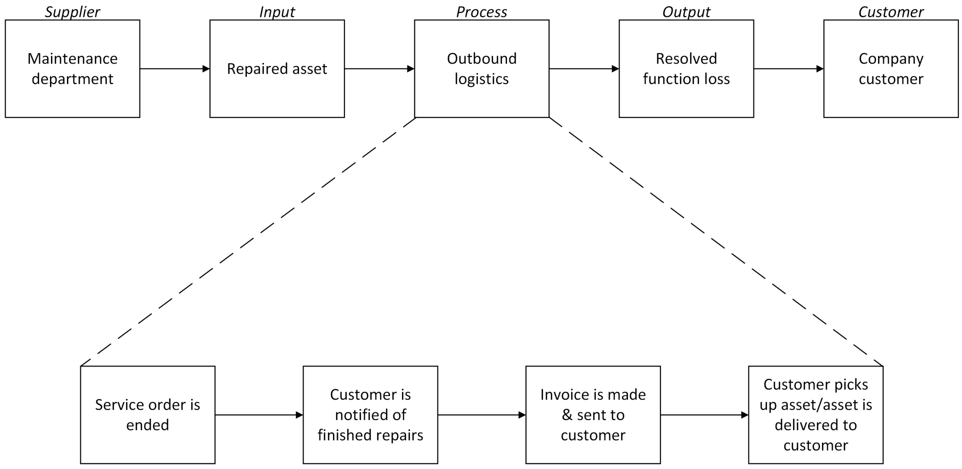


Figure 6: Outbound logistics

### 2.1.1 Service Process

The whole service process is also analysed; this process entails all facets that happen whilst the equipment is in for servicing. To analyse this process a Business Process Model and Notation (BPMN) was chosen. A BPMN is a standardised graphical language used for modelling and describing business processes in a clear and understandable way. [6] This method is chosen due to its easily understandable notation.

The servicing process starts when the customer requires service. To begin the process, the customer contacts customer service to say they have equipment that needs maintenance. This contact can be done via telephone, face-to-face, or e-mail. The customer service employee will make a service order for one (or multiple) pieces of equipment. After finishing the service order, the customer service employee sends the service order to the maintenance department. During this process, it also must be considered if the equipment is brought to the maintenance department or needs to be picked up by Company X' logistics department. If the last is the case, a sub-process is started, which sends the logistics department an assignment to schedule and pick up the particular equipment(s). The maintenance department mainly considers the service orders on a First Come, First Served (FCFS) basis, meaning that an incoming service order ends up at the bottom of the pile awaiting servicing. However, in reality, some service orders receive priority. This happens due to several reasons:

- The service order is from a frequent customer, therefore being more valuable and receiving a higher priority. Frequent customers are mostly corporate customers, like landscaping companies.
- A relatively small service order comes in, and the maintenance employee decides to do that before continuing with the service orders awaiting maintenance.
- A service order may have a higher urgency than another one. For example, lawnmower service orders need not be serviced urgently during colder months.
- Mechanic misbehaviour; a mechanic may not 'enjoy' working on a certain job, therefore leaving it and doing other tasks first.

When a mechanic starts working on a service order, the first step is to inspect the equipment to identify any issues or problems. This is a vital step in the process, as the service order does not always depict the problem that must be fixed. The mechanic can resolve the issue immediately and complete the service order if the required materials and spare parts are available. However, the necessary materials and spare parts must be ordered if unavailable. The mechanic requested that the Sales/Purchasing department order the spare parts. During this time, the service order will be put on hold. Once the spare part(s) arrive at the logistics department, the department confirms the order and ensures that the spare part(s) are perceived to be in stock. This allows the mechanic to continue with the service order. Once the mechanic receives the spare parts, they can complete the repair and service order. The customer can then collect the equipment or transport it to the customer. To complete the process, the service order must be ended, and the customer service will send an invoice for the service order.

#### BPMN Description

Figure 7 shows the Business Process Model and Notation (BPMN) for the process sketched in Section 2.1.1. The process starts in the **customer** swimming pool; the starting event is a maintenance need of the customer side. To do something about this need, the customer will contact the customer service department, with whom the customer will discuss their maintenance needs.

This starts a sub-process in the **customer service** swimming lane, during which the **customer service** produces a service order for the discussed needs. This service order will be sent to the **maintenance** swimming lane. After discussing the maintenance need with customer service, the **customer** should decide if they bring the equipment(s) in themselves; if this is the case, the primary process continues to the **maintenance department** swimming lane. If this isn't the case, another sub-process in the **logistics** swimming lane is started to bring the equipment(s) to the **maintenance** department. In this sub-process, the **logistics** department schedules a pick-up moment with the **customer**, followed by the actual pick-up. And to conclude the sub-process in the **logistics** swimming lane, the equipment is dropped off at the **maintenance** department.

When the service order and equipment(s) arrive in the **maintenance** swimming lane, the service order can be scheduled. As mentioned in Section 2.1.1, the service orders are mainly scheduled following an FCFS basis. The mechanic starts working on the equipment by inspecting it first, after which the mechanic asks if spare parts or materials are available to fix the problem. If this is the case, the mechanic can resolve the maintenance immediately and solve the issue directly. If this isn't the case, another sub-process will be started. This sub-process is created in the **maintenance** swimming lane, where the mechanic will notify the **sales & purchasing** department to order the required spare parts or materials. The mechanic will also put the service order on hold. This sub-process continues from the **sales & purchasing** swimming lane to the **logistics** swimming lane, which will receive the ordered materials. The **logistics** department will move the spare parts and materials to inventory from where the **maintenance** department can pick them up.

After picking up the spare parts, the mechanic can perform the maintenance and finish the service order. Lastly, a service invoice must be sent to the customer by the **customer service** department. After which, the customer can pick up the repaired equipment, or the **logistics** department schedules the fixed equipment(s) for delivery.

## 2.2 Current Scheduling Method

The current servicing process follows a First-Come-First-Served (FCFS) basis for scheduling, where the orders are placed at the back of the queue and serviced accordingly. However, some customers, such as business customers like landscaping companies, are given priority over private customers. This is because of the significance these companies hold regarding efficient equipment usage. Service times

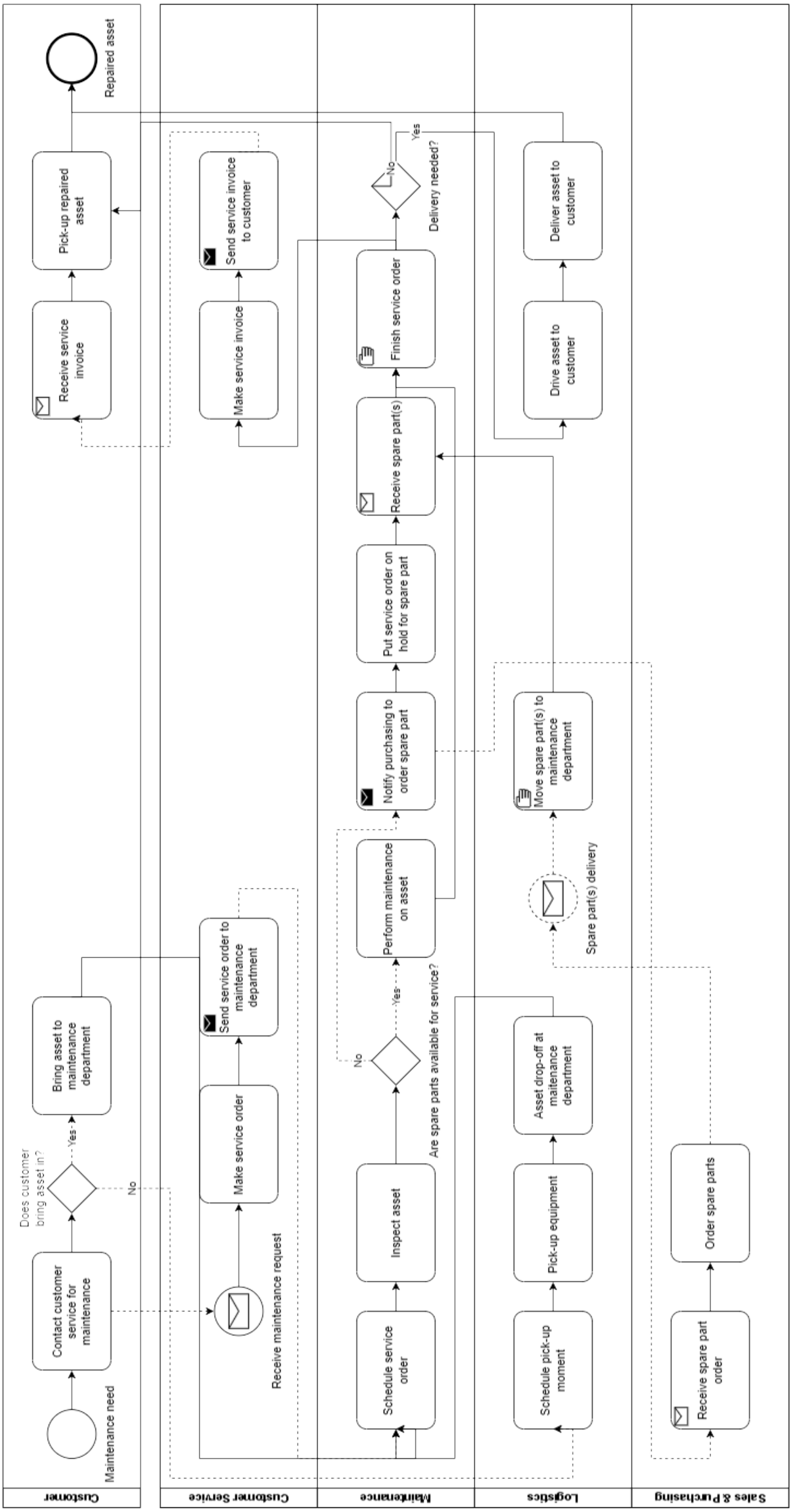


Figure 7: BPMN Service process

are not considered; deadlines are only considered when customers request their equipment. However, based on the employee interviews, the queue management does not always follow the expected protocol. This causes a few inefficiencies in the servicing process:

- Customers go directly to the maintenance department, and mechanics directly take small service instances, keeping them from the scheduled jobs.
- Mechanics assign their 'own' priority to the scheduled jobs, as they don't feel like doing a particular service order.

## 2.3 Data Analysis

Data analysis plays a vital role in obtaining information for evidence-based decision-making. It involves collecting and examining data to turn it into informative insights. The primary purpose of this analysis is to gather information related to operations, service instances and service times, which will serve as input for the scheduling method.

To develop a well-functioning job scheduling method, reliable and complete historical data related to past operations is required. This data is gathered during the steps outlined in section 2.1. The customer service employee collects general job data such as defect description and service order data, while the maintenance department provides job-specific data like performed maintenance and used materials. This data is stored in the ERP system and can be viewed in Figure 8.

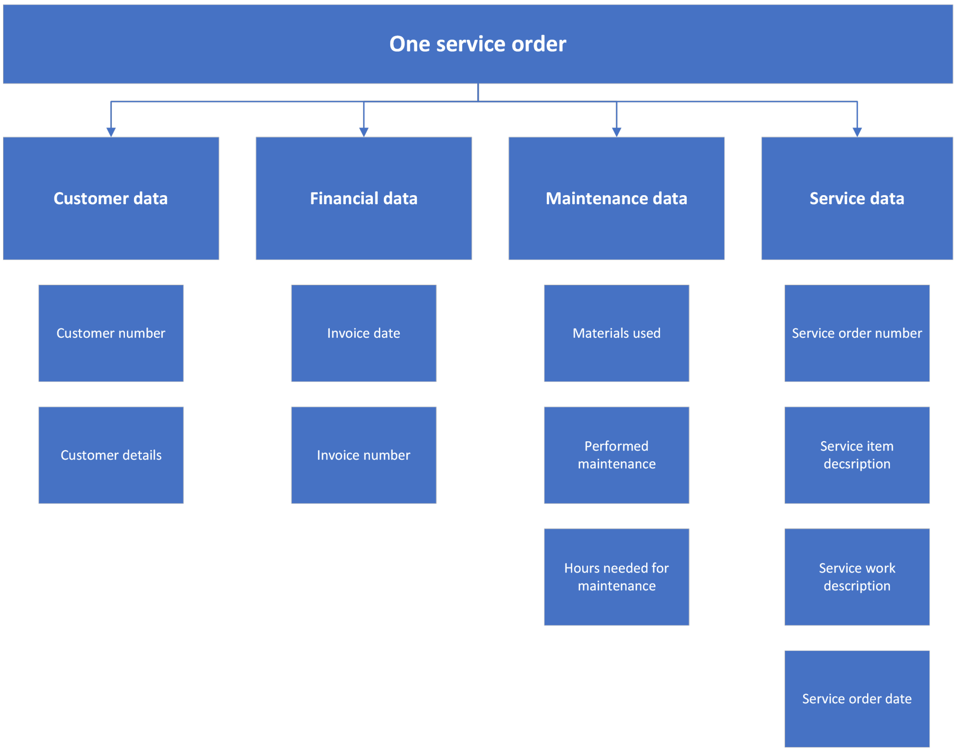


Figure 8: Available data

The past two years of performed operations are available for research, 2022 and 2023. These years are believed to be the most valuable data the ERP system provides, as older years do not provide a valid reflection of reality, operations-wise. The considered data is directly from the company's ERP system. Therefore, it is believed to be reliable to use in this research. Only the data from 2023 will be used for analysing the servicing process, as this year's data is deemed to be the most valuable for this research. This year's data is used as Company X has had recent operational development, mainly considering the daily operations in the workshop, as will be discussed in Section 2.4.2.

### 2.3.1 Suitable Data

Proper categorisation of data is vital to ensure its usefulness. Given the diverse range of equipment handled by the maintenance department, multiple categories that serve as the basis of the scheduling methodology are devised. Furthermore, every service instance will be allocated a category for implementation. However, it is essential to note that not all data within these categories can provide valuable insights for the scheduling model. A list of preferred data inputs most suitable for inputting into the job scheduling model has been curated to address this. The list is as follows:

- Categorised service occurrences, an overview of equipment categories with their corresponding number of service occurrences during the given period.
- Average service time, distributed per category, the average processing times were in the given period.
- Service time distribution, per category, is the upper and lower bound of the processing time during the given period.
- Workshop capacity, registered hours for the availability of mechanics.

As most data is registered manually in the ERP system, some data must be interpreted, and others must be omitted. For example, the mechanics manually input the materials and work performed into the ERP system. Therefore, this data is subject to human error and also needs interpretation. The entire dataset is validated to remove these errors, and all raw data is ensured to be valid. Each service instance is provided with a preliminary category to get a first overview of the categories.

As previously stated, the service data was initially sorted into various equipment categories. However, these categories proved too broad for usage, resulting in 68 categories. Therefore, a manual selection was made to remove unnecessary categories and merge some categories to result in a more usable number of categories. This process formulated a list of categories, as shown in Table 1.

Workshop capacity is among the most crucial data for developing a maintenance schedule, as this data directly influences the days maintenance can be performed. Therefore, historical data will be gathered on the days that the mechanics weren't present in the workshop and, therefore, couldn't perform any maintenance. Examples of these days are weekend days, public holidays, and days when one or more mechanics perform on-site maintenance. This will be used to map the mechanic availability and, thereby, the workshop capacity.

### 2.3.2 Service Occurrences

The equipment categories are used to search for and group all the service orders into relevant categories. As illustrated in Figure 9, the service occurrences reveal that chainsaws, lawnmowers, and lawn tractors are some of the primary categories that the maintenance department services. These categories are responsible for a significant part of the company's overall service operations, mainly because they are its main selling point. Therefore, they contribute to a large proportion of the maintenance operations.

### 2.3.3 Service Times

A study was conducted to analyse the service times through a comprehensive data analysis and several employee interviews. Informal interviews were conducted with the mechanics to understand service times' administration better. The findings revealed that the administration of service times is composed of two elements. Firstly, the frequently occurring maintenance issues are categorised

Table 1: Equipment Categories

| Category | Description |
|---|---|
| Brushcutter | Machine used for trimming weeds, small trees and other foliage, especially for hard-to-reach places |
| Chainsaw | Motorised portable used for cutting trees |
| Engine | A machine that uses the energy from liquid fuel or steam to move |
| Generator | Motorised electricity generator, used for providing electricity in areas without regular electricity |
| Grinder | A device with a rotating abrasive disc used to grind, polish, or cut metal and other materials |
| Hedge trimmer | Machine used for trimming hedges |
| KombiEngine | Motorised tool used for landscaping purposes |
| Lawnmower | A machine used for cutting grass |
| Lawn scarifier | A tool with sharp points used for breaking up and removing unwanted grass or plants that are covering the surface of a lawn |
| Lawn tractor | Tractor used for mowing large sections of grass, not achievable with a regular lawnmower |
| Leaf blower | A piece of equipment for clearing leaves that have fallen from trees by blowing them away |
| Other | Diverse (gardening) equipment which couldn't be used in another category |
| Plate compactor | A machine that presses down on soft ground, compacting and preparing for pavement |
| Pressure washer | A machine that sends out a strong beam of water, used for cleaning things |
| Pruning machine | Industrial machine used for trimming down plants |
| Robotic lawnmower | Robotized lawn mower |
| Saw | tool used to cut down trees |
| Sweeper | Industrial machine used for sweeping big floorspaces |
| Trailer | Car attachment used to transport goods |
| Waterpump | Heavy-duty pump used to move large amounts of non-drink water |
| Weed brushing machine | Machine used to get rid of weeds stuck in between pavement |

into standard service orders for which standard times are established for the mechanic to perform the service. Secondly, additional time is required to complete the service in some cases. For instance, when a customer brings their lawnmower in for winter maintenance, it is considered a standard service order with a pre-specified service time. However, the mechanic may discover a defect that requires fixing and takes longer than the pre-specified service time. In such cases, the mechanic documents the additional service performed and the time it took to repair the defect. As a result, the service time of a single service order is comprised of two elements, as illustrated in Figure 10.

Two different approaches were used to analyse these two elements. The standard service orders were analysed through employee interviews with the mechanics. Their expertise was used to understand the time required for a single standard service order. These standard service orders are categorised, and their service times are shown in Table 2. The (extra) service time for service orders is analysed using a dataset extracted from the Company X ERP system.

The dataset provided in this context classifies the duration of extra time that a mechanic spends on a single service order. This additional time is solely dedicated to maintaining a single piece of equipment and does not include any setup or changeover time required while switching between service orders. The times displayed in Figure 11 do not consider the times mentioned in Table 2. It is important to note that a single service order can consist of both standard service time and extra service time. The orange line in the graph depicts the average *extra* service time for each category, with the average processing time of a single service order being 57 minutes. Table 3 shows descriptive statistics regarding each equipment category, showing mean processing time, maximum processing time and minimum processing time in hours, service occurrences and mean processing time in minutes.

The graph indicates that some categories, such as brushcutters and chainsaws, have a lower average processing time, while others, such as lawn tractors and trailers, have a higher one. There could be various reasons, but equipment like brushcutters and chainsaws are assumed to be less complex than lawn tractors. Therefore, expecting these categories to have lower processing time averages is reasonable.

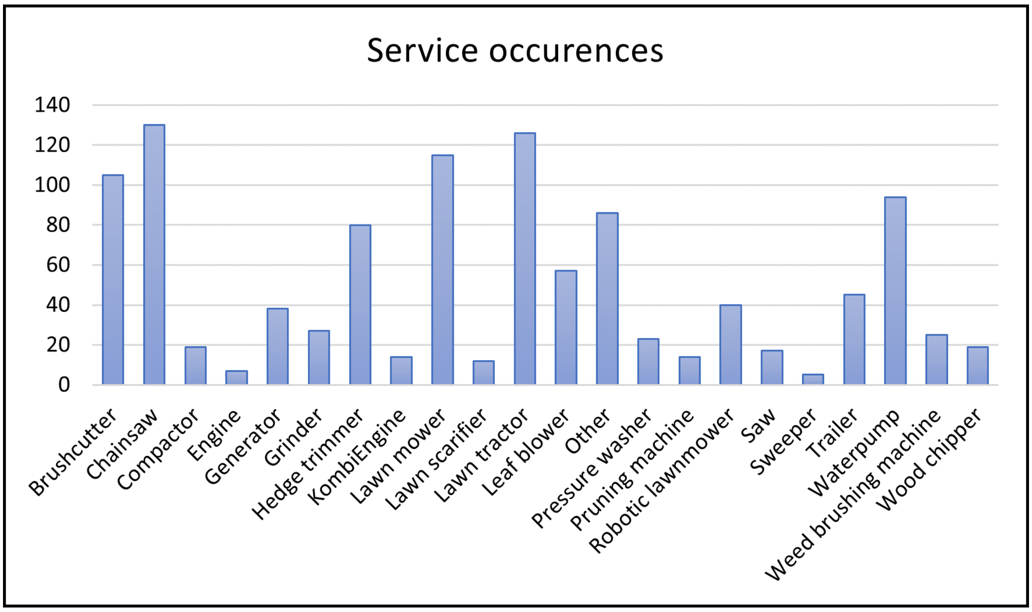


Figure 9: Categorized service occurrences

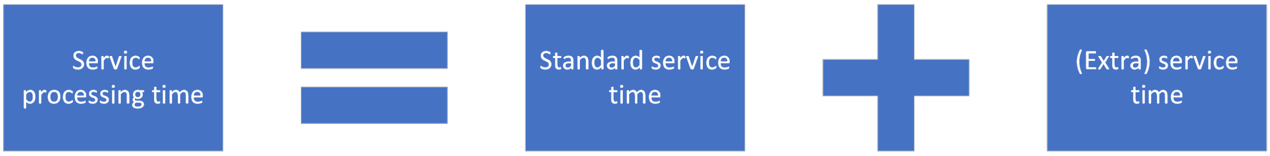


Figure 10: Service time set-up

### 2.4 Scheduling Characteristics & Requirements

The current scheduling and maintenance processes have been analysed. As the maintenance department handles many different service orders, various characteristics are identified to formulate the scheduling method properly. These and other essential characteristics, identified via discussions with Company X employees, are elaborated on further. These characteristics are seen as requirements for the scheduling method.

- Job category
- Workshop capacity
- Job duration

#### 2.4.1 Job Category

A job category describes jobs entering the system. These job categories are directly taken from the equipment categories, as shown in Section 2.3.1. These categories showed to have a widely varying average service time (see Section 2.3.3), which will be used as a starting point for the input of the scheduling model. Therefore, the solution model should consider these categories.

#### 2.4.2 Workshop Capacity

The maintenance schedule's capacity can differ per day in a given week. The mechanics sometimes perform on-site maintenance, so they cannot work in the workshop. This work is also not considered for the scheduling model. Therefore, when a mechanic is not present at the workshop, their working

Table 2: Standard Service Orders and Service Times

| Jobcode | Description | Processing times (h) |
|---|---|---|
| R0070 | Lawnmower maintenance (electric) | 0.75 |
| R0080 | Lawnmower maintenance (petrol) | 1.00 |
| R0090 | Lawnmower maintenance (driven) | 1.25 |
| R0100 | Lawn tractor maintenance | 2.00 |
| R0130 | Sharpening blades lawn tractor/lawnmower | 0.25 |
| R0140 | Sharpening blades (big) | 0.25 |
| R0150 | Sharpening & maintenance robotic mower | 1.50 |
| R0160 | Sharpening hedge trimmer (petrol) | 0.50 |
| R0190 | Replace cutting line brushcutter | 0.10 |
| R0230 | Wheel balancing | 0.30 |
| R0270 | Inspection inc. Sticker | 0.25 |
| R0290 | Sharpening reel mower | 1.25 |
| R0300 | Sharpening hedge trimmer (electric) | 0.50 |
| R0310 | Sharpening cutting blade inc. Assembly | 0.25 |
| R0320 | Sharpening hedge trimmer | 0.20 |
| R0330 | Sharpening pruners | 0.20 |
| R0450 | Start cord replacement | 0.20 |
| R0510 | Sharpening chainsaw | 0.10 |

hours shouldn't be considered, which means that the workshop capacity on the given day should be reduced. This mechanic availability implies that a mechanic works an 8-hour shift (or 480 minutes). During a given week, a mechanic can have at most 5 days at the workshop, where each day there is at least one mechanic present in the workshop. Which depends on the demand for on-site maintenance.

#### 2.4.3 Job Duration

Directly defined from the job category scheduling characteristic, the duration of a single job instance must also be included in the solution model. This processing time will be derived from the input from the data analysis in Section 2.3.

The processing time for a single service order entails two components, the standard servicing time and (possible) extra service time, as mentioned in Figure 10. Therefore, the scheduling model should consider both components when scheduling the jobs.

### 2.5 Assumptions

Several assumptions have been made to ensure the effective formulation of a scheduling method.

#### 2.5.1 Release Dates

All jobs for a given week are scheduled at the beginning of next week, assuming the maintenance department receives all jobs on the first day. This day is considered the release date of the service

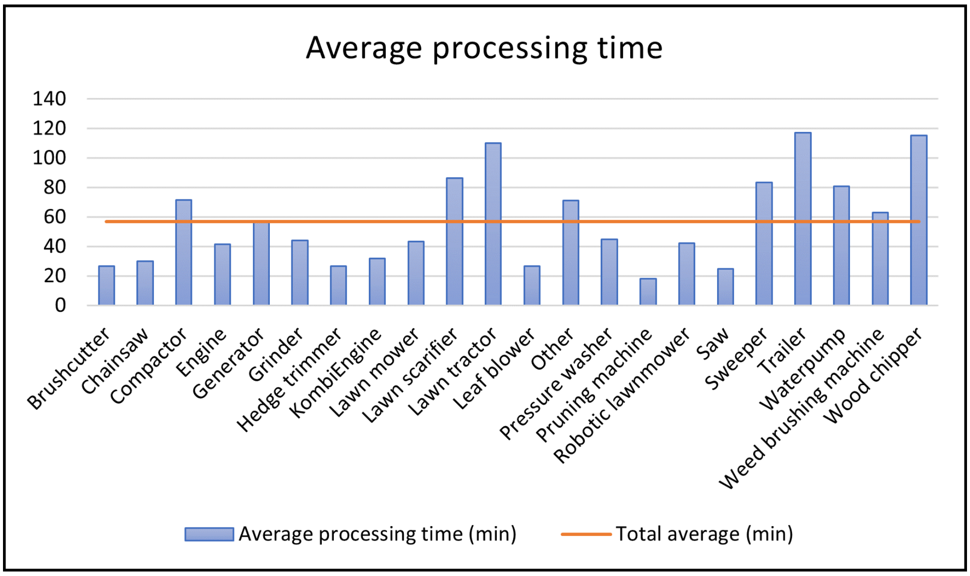


Figure 11: Average processing times

order.

#### 2.5.2 Processing Time

The time taken to process a particular job depends on either one or both factors described in Section 2.4. Additionally, the processing time may involve various standard servicing types. If only one of the two factors is considered, the other is assumed to be zero.

#### 2.5.3 Changeover Time

Changeover time, the time taken from the end of a job to the start of the next job, is assumed to have a mean of 5 minutes. This 5 minutes was decided upon, as from the workshop interviews arose that the time in between two service orders wasn't considered in the time registered, and the mechanics agreed that it would be 5 minutes on average. This changeover is the time from the successful completion of one job to the start of the following job; administration is considered in the service time.

### 2.6 Constraints

Constraints must be set to develop an effective scheduling method for the maintenance department. These constraints will play a pivotal role in modelling the solution method, which will be discussed in Section 4.

#### 2.6.1 Working Hours

A day has at most eight working hours, and a week has five working days. Therefore, a mechanic can work 40 hours per week at most. As there are three mechanics, the maximum capacity of the workshop is set to 120 hours per week. Mechanics can also perform service on-site. If maintenance is performed on-site, one or two mechanics will go to the worksite and cannot perform any service in the workshop that day. The third mechanic will work in the workshop independently, bringing the workshop's capacity to 8 hours for that day. This can happen any day; therefore, the workshop's capacity is between 40 and 120 hours.

Table 3: Categorised Service Statistics

| Category | Mean processing time (h) | Maximum (h) | Minimum (h) | Mean processing time (min) |
|---|---|---|---|---|
| Brushcutter | 0.44397 | 1.50 | 0.10 | 27 |
| Chainsaw | 0.49679 | 3.00 | 0.10 | 30 |
| Compactor | 1.19211 | 3.50 | 0.20 | 72 |
| Engine | 0.69286 | 1.25 | 0.25 | 42 |
| Generator | 0.95658 | 2.50 | 0.25 | 57 |
| Grinder | 0.73272 | 3.00 | 0.20 | 44 |
| Hedge trimmer | 0.44667 | 2.00 | 0.10 | 27 |
| KombiEngine | 0.53095 | 1.50 | 0.20 | 32 |
| Lawn mower | 0.72246 | 4.00 | 0.10 | 43 |
| Lawn scarifier | 1.44167 | 9.25 | 0.25 | 87 |
| Lawn tractor | 1.83532 | 11.00 | 0.10 | 110 |
| Leaf blower | 0.44649 | 1.50 | 0.25 | 27 |
| Other | 1.18547 | 9.00 | 0.10 | 71 |
| Pressure washer | 0.74710 | 4.70 | 0.10 | 45 |
| Pruning machine | 0.30238 | 1.00 | 0.10 | 18 |
| Robotic lawnmower | 0.70250 | 3.50 | 0.10 | 42 |
| Saw | 0.41176 | 1.00 | 0.25 | 25 |
| Sweeper | 1.39000 | 2.50 | 0.75 | 83 |
| Trailer | 1.95333 | 25.00 | 0.25 | 117 |
| Waterpump | 1.34840 | 4.50 | 0.25 | 81 |
| Weed brushing machine | 1.05200 | 2.80 | 0.25 | 63 |
| Wood chipper | 1.92105 | 7.00 | 0.30 | 115 |

#### 2.6.2 Overtime

The schedule doesn't consider overtime; a job can only be entered into a given day's schedule if time is still left. Jobs cannot be scheduled if there isn't enough time in the system, and jobs can not be split over two days. The next day is attempted; if the job cannot be scheduled on the current day, the next day is attempted.

#### 2.6.3 Priority

Some jobs are given priority over others, depending on specific criteria. Jobs left over from the week before are given priority over all other jobs and are scheduled at the start of the following week. This is done as there was no space left in the schedule from the week before. If no jobs are left over from the week before, another priority is given to commercial clients, like landscaping companies. These companies heavily rely on their equipment functioning, and therefore, these clients receive priority before all other priority rules (except for the leftovers from last week).

### 2.7 Key Performance Indicators

Key Performance Indicators (KPIs) are measuring instruments used to check the efficiency and specific processes. For this scheduling method, two KPIs are determined by the schedule's performance with the proposed methods.

#### 2.7.1 Total Makespan

The total completion time is needed to determine the solution model's efficiency. The total makespan in the solution model is the duration from initiating the first job to completing the last job, representing the overall time required to execute all jobs in the schedule. This total makespan is very useful for quickly comparing schedules, as it means the whole time taken for the schedule to complete all jobs.

### 2.7.2 Average Daily Idle Time per Mechanic

Another critical performance indicator is idle time when a mechanic is not working on a service order. This time does not include changeover time between jobs but only contains the time that a mechanic is available for work but is not or cannot be assigned a job. The average daily idle time per mechanic incorporates the overall idle time for the whole schedule.

### 2.7.3 Efficiency Ratio

The efficiency ratio is measured by how much time resources are converted into output for the maintenance department. This ratio is calculated by determining how much time is spent on processing jobs and how much time is spent idle in the system, resulting in a percentage of time. Changeover time is considered processing time as it is necessary for the workshop.

### 2.7.4 Computation Time

Lastly, the computation time of the heuristic will be considered. As many calculations and evaluations will be done, the scheduling method can take some time to find an improved solution. Therefore, the computation time will be used as a KPI.

## 2.8 Current Scheduling Method

The current scheduling methodology is based upon the First Come, First Served (FCFS) principle. The first job entering the process is served as first, and the second job is second, etc. However, during conversations with employees, it arose that not all jobs follow this principle. Jobs from landscaping companies receive priority as they rely heavily on their equipment functioning. Therefore, these jobs receive priority. The inefficiencies mentioned in Section 2.2 can also be seen as (unwanted) priorities; however, these occurrences' randomness cannot be modelled and aren't considered.

This scheduling method will be used to test it against other scheduling heuristics. And therefore is modelled in a step-by-step approach:

1. Order to-be scheduled jobs in chronological order (based on upon service order dates) for the specified timeframe (month or week)
2. Starting a t = 0, schedule first job
3. If a mechanic is free, schedule the next job
4. Update t
5. If jobs are still to be scheduled, return to step 3
6. Otherwise finish schedule, update total processing time = t
7. End schedule

## 2.9 Summary

The context analysis is formulated to understand the current situation Company X operation is handling comprehensively. The whole process is analysed, and a BPMN is designed for servicing. The current scheduling method is analysed and will be used to model the current situation in the solution design phase. Furthermore, the service times are examined, and the scheduling characteristics, requirements, constraints, assumptions and KPIs are formulated. All are contributing to answering the research question of this chapter:

*"How is Company X currently handling their service operations, and what are the service times associated with these operations?"*

#### Current service operations

The current service operations are handled based on a first in, first out principle; service orders are based on arrival date in the servicing department. The current service operations are analysed through observation and employee interviews, which resulted in the detailed BPM of Figure 7.

#### Service times

The service times are analysed, and it was found that the service time consists of two aspects (not including changeover time): standard service jobs and extra service time. The standard service jobs are analysed through the mechanic interviews, which resulted in the standard service times of Table 2. The extra service time is incurred if the maintenance is not included in the standard service jobs. These times are analysed by assigning categories (shown in Table 1) to each of the service jobs; this is done as the jobs are considered to be of a wide variety, and this way, the jobs can be analysed based on these categories. The associated service times for these categories can be found in Figure 10 and Table 3

# 3 Theoretical Framework

This chapter will provide a theoretical framework for formulating the scheduling method. Section 3.1 will discuss the general theory behind job scheduling. Then, Section 3.2 will introduce different scheduling problems and Section 3.3 will introduce the knapsack problem and discuss its relevance to the situation at the maintenance department. Section 3.4 introduces heuristics, which is a way to solve job scheduling problems. Next, Section 3.5 will identify the scheduling problem best applicable to Company X. Then, Section 3.6 will discuss case studies containing similar situations to the maintenance department. Lastly, Section 3.7 will summarise the chapter and answer the research question for this chapter:

*What relevant theories best apply to the maintenance department to develop a schedule improvement method?*

## 3.1 Scheduling

In everyday life, scheduling is encountered in all walks of life. Primarily in public spaces, schedules are encountered. For example, when using public transportation, schedules ensure that the means of transport arrive and depart on time. Kenneth R. Baker formulated [7] this process of making a schedule allocating resources over time to perform a collection of tasks. Scheduling primarily took an interest in the manufacturing industry at first, as problems faced in this industry were first taken into account by scheduling theory. For this reason, scheduling terms used for resources and tasks are called machines and jobs, respectively. Nowadays, scheduling theory is used in various sectors as it has proven to contribute significantly. [8]

This scheduling theory is primarily based on using mathematical models to create schedules. These models aim to convert the problem structure into a mathematical form. Baker and Trietsch have defined this quantitative approach as the "description of resources and tasks with the translation of decision-making goals into an explicit objective function" [8]. The objective function can be regarded as the primary goal of the scheduling problem. It should encompass all costs that depend on scheduling decisions. However, due to complexity constraints, this isn't easy to measure. Therefore, three decision-making types have been identified: turnaround, timeliness, and throughput. Turnaround is the time taken to complete a task. Timeliness measures if a particular task can be completed within a specified deadline, and throughput measures the work completed in a specific amount of time [8].

Several notations are used to translate scheduling problems into a mathematical model. These notations form the basic information regarding to-be scheduled tasks and provide the necessary details in scheduling theory. The collection of tasks is identified as a set of $n$ jobs $J_i$ $(i = 1, ..., n)$, which can be processed on a set of $m$ machines $M_j$ $(j = 1, ..., m)$. Throughout scheduling research, it is assumed that a machine can process at most one job at a time, and each job can be processed by one machine at a time. [9] Regarding a job, a lot of information can be gathered. The following information is widely used when working with job scheduling [9]:

- The number of operations $n_i$
- Processing time $p_{ij}$, that job $i$ has to spend on the machine $M_j$
- Release date $r_i$ denotes the time job $i$ becomes available for processing
- Due date $d_i$, the time job $i$ should preferably be completed
- Weight $w_i$, a number for the relative importance of job $i$

A widely used way of presenting a job schedule is through a Gantt chart. This chart visually represents the machines and jobs involved in a scheduling problem and how they are scheduled over time. Figure 12 shows an example of a single-machine schedule. In this schedule, a single machine processes three jobs sequentially.

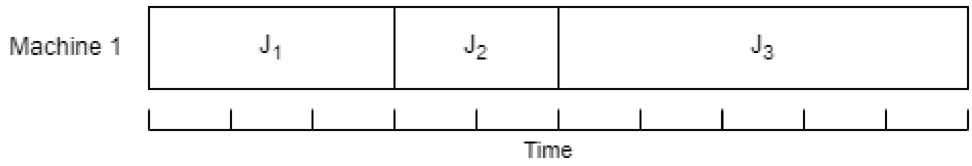


Figure 12: Example of a schedule

### 3.2 Scheduling Problems

In scheduling theory, "scheduling problems" refers to a broad problem category. As discussed in Section 3.1, job scheduling has different characteristics. A scheduling problem is an example of a job scheduling assignment. In this section, we will delve deeper into the characteristics of scheduling problems. Graham et al. researched these characteristics and proposed a categorised notation for scheduling problems [9]. This notation includes the terms $\alpha, \beta$ *and* $\gamma$. $\alpha$ represents the machine environment, $\beta$ represents job characteristics and constraints, and $\gamma$ represents the objective function.

#### 3.2.1 Machine Environment

The machine environment refers to the conditions the scheduling problem must adhere to regarding the different jobs. It can be seen as a machine configuration to which the jobs are subject. When dealing with multiple machines, a job can consist of multiple operations each machine should process. The machine environment can be classified as either a single-stage environment, where each job has only one operation, or multi-stage machine systems, where each job has multiple operations and different stages.[10]

In a single-stage system, multiple machines can also be considered. In this case, there are three types of machine environments: identical parallel, uniform parallel, and unrelated parallel machines. Identical parallel machines have the same processing time for each machine. This means that the processing time of a job is independent of which machine it is processed on. Uniform parallel machines are the same as each other, except for the processing times. Each machine will run at a different speed. In contrast, unrelated parallel machines have processing times that depend on the machine.[10] An example of a single-stage multiple-machine schedule is shown in Figure 13.


Figure 13: Example of single-stage multiple machine schedule

Graham et al. [9] identified different multi-stage machine environments. They discuss problems where jobs are required to be executed by multiple machines, giving each job various operations. These problems are open shop, flow shop, and job shop problems, which are all multi-machine scheduling problems. Open shop problems are scheduling problems in which the order of job execution is irrelevant, and jobs can pass through the different machines in any random way possible. Flow shop problems have a selected order in which the jobs pass through the machines, and every job passes along the machines in the same sequence. Lastly, job shop problems must pass along all machines, but each job category can have a different sequence specification. Figure 14 shows an example of a multi-stage schedule. This schedule shows a three-machine flow shop scheduling problem.

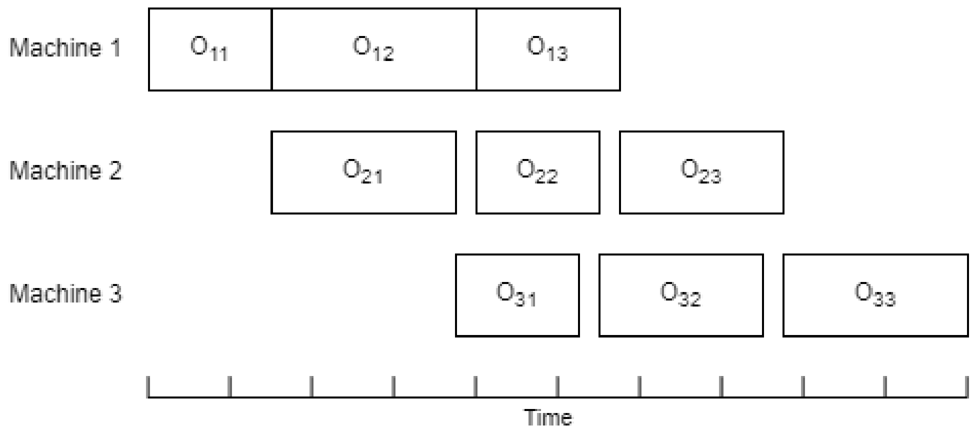


Figure 14: Example of multi-stage schedule

### 3.2.2 Job Characteristics & Constraints

Job characteristics are of significant importance in defining the scheduling problem to be dealt with. A job's first, arguably the most important, characteristic is its processing requirements. This considers the time taken to process a job. Other job requirements are characterised by the availability for processing, dependence on preceding jobs and whether interruptions during processing are allowed. [10]

Jobs which are characterised by availability have a specified release date. This means that each job gets an assigned integer value, which denotes when a particular job becomes available for processing and can enter the schedule. A job can also be characterised by a pre-specified deadline; these deadlines are used to prioritise specific jobs. The use of these deadlines will be elaborated on in Section 3.4.

Precedence constraints can also characterise a job. Precedence constraints are a form of job dependence for which a certain job depends on completing another job. This means that if a job $j$ has precedence over job $k$, it can only start job $k$ if job $j$ is completed. [10]

Lastly, the job can be characterised by the possibility of interruptions. This is called *preemption* regarding scheduling problems, meaning that the job can be interrupted if there is a need for it, and the operations can be resumed at a later time. If jobs are non-preemptive, it is not possible to interrupt a job that is being processed. [10]

### 3.2.3 Objective Function

The objective function is the main objective in classifying a scheduling problem. This main objective follows from the desire when working with a scheduling problem. A scheduling problem can be developed to decrease total completion time ($C_{max}$), also known as the *makespan* [10]. Other objective functions, like the maximum lateness (defined as $L_{max}$). Remember the scheduling goals from Section 3.1; each objective function can be classified in one of those scheduling goals. For example, the timeliness goal categorises an objective function using the maximum lateness.

### 3.2.4 Relevance to Workshop Schedule

Job scheduling or (job) scheduling problems lend their usefulness to the application of job allocation to a variety of resources. The workshop schedule has a good resemblance to the typical scheduling problem, especially regarding the machine environment and job characteristics. The workshop schedule's machine environment can be depicted as three machines (i.e. the mechanics), which work in parallel in the workshop. The job characteristics also resemble the scheduling problem, as a scheduling problem often has deadlines, release dates, or other job characteristics that should be considered when making a schedule. This is also the case for the workshop schedule, as the workshop employs release date and preemption constraints.

## 3.3 Knapsack Problem

Next to the theory of scheduling problems, relevant theory can also be identified in the form of other optimisation problems. These relevant theories are identified as they relate to the situation at the maintenance department; for this, the knapsack problem is identified.

The knapsack problem is a combinatorial optimisation problem. This optimisation problem is named after the analogy of the knapsack, which is a bag carried on the back or over the shoulder. This knapsack has a finite capacity; once it's full, nothing else can fit in it. This can be represented by a set of entities with a corresponding value, and the value cannot exceed a prefixed bound. These entities are called items $j = \{1, 2, ..., N\}$ for a given set of items $N$, and each item has a profit $p_j$ and weight $w_j$. [11]

The most used knapsack problem is the 0-1 knapsack problem [12]. This knapsack problem deals with the most basic form of filling a 'knapsack' until the sum of weights of the assigned items reaches a prefixed weight $W$. This equation maximises its objective function under the constraint of not exceeding the value of $W$ and is formulated as follows:

$$max \sum_{i=1}^{n} p_i x_i \quad \text{subject to} \quad \sum_{i=1}^{n} w_i x_i \leq W \quad with \quad x_i \in \{0, 1\}[12] \tag{1}$$

Many variants of the original knapsack problem can be identified and developed in recent years [12], each having different constraints and characteristics. The knapsack problems with item types consider multiple identical copies of each item, referred to as item types. This allows for more detailed modelling of real-world scenarios where items are not unique, but numerous copies exist. Three variants of knapsack problems with identical item types are considered: the Bounded Knapsack Problem (BKP), Unbounded Knapsack Problem (UKP) and Change-making problems. For the BKP, there is a bounded number of identical items, and for the UKP, there is an unlimited supply of copies. Lastly, change-making problems consider selecting items to meet a specific target value or weight. [12]

Two other variants of the original knapsack problem are the Knapsack Problem with Setup (KPS) and the Multiple-Choice Knapsack Problem (MCKP). The KPS, as the name suggests, considers setup costs for the items that need to be placed into the knapsack; thanks to these setup costs associated with selecting items for the knapsack, more realistic representations of real-world scenarios can be modelled. Furthermore, the MCKP forces deviate from the original 0-1 Knapsack Problem by assigning classes to items; for each of these classes, one item must be selected and, therefore, also helps to model real-world scenarios. [12]

A knapsack is mostly formulated as an Integer Linear Programming (ILP) problem (as shown in Equation 1) and, therefore, can be solved using exact algorithms. [11] ILP allows Knapsack Problems to be solved by an ILP solver; these solvers are very useful in finding the optimal solutions for ILP problems. However, they are extremely computationally heavy. Therefore, ILP can only be solved for relatively small problems. [13] Other exact algorithms that can be used for solving knapsack problems are Dynamic Programming (DP) and Branch-and-Bound. DP involves breaking down the main problem into smaller sub-problems and solving these recursively; the solutions are stored in a table and used to solve larger sub-problems until the optimal solution is found. The Branch-and-Bound method systematically explores the solution space by balancing each decision point and bounding under certain criteria. Again, both of these methods are computationally intensive. [11]

### 3.3.1 Relevance to Workshop Schedule

The relevance of knapsack problems towards developing a service schedule for the maintenance department can be seen when looking at the service schedule on a lower scale. Each working week has five days, and during a day, at least one and at most three mechanics are working in the workshop. Each of these mechanics has a finite capacity of eight working hours for which service jobs can be scheduled. It doesn't matter in which sequence these jobs for a mechanic are ordered as they have no sequence setup times, and the mechanic has a preset processing time for a selected job. Therefore, the capacity of the mechanic can be seen as the capacity of the knapsack, and the service times associated with the available jobs can be seen as the weights of the items that can be selected for the knapsack problem.

## 3.4 Heuristics for Scheduling Problems

Several approaches, like exact algorithms, can solve these job scheduling problems. Exact algorithms produce an optimal solution for the scheduling problem. However, exact algorithms can only solve relatively small scheduling problems [14]. Therefore, heuristics are employed in practical scheduling problems. These heuristics are ways to solve and compute various problems computationally; however, using heuristics does not indicate how close the solution is to the optimal situation. [13] The following section will elaborate on different heuristics and determine their effectiveness in scheduling problems

### 3.4.1 Rule Based Heuristics

Heuristics are methods used to prioritise jobs and schedule them accordingly. These prioritisation methods are always based on the objective function discussed in Section 3.2.3. One well-known priority rule is the Earliest Due Date (EDD) priority rule. This rule arranges the jobs in order of their due dates, resulting in a schedule with increasing due dates. This is the most fundamental form of scheduling as it is straightforward. If no due dates are assigned to the jobs, they can be prioritised based on their processing time. Examples of such heuristics are the Shortest Processing Time (SPT) and Longest Processing Time (LPT). These heuristics use the processing time to assign priority. When employing these priority rules, the jobs are sorted based on processing time and scheduled in this manner, picking the shortest and longest processing time, respectively. [15]

### 3.4.2 Apparent Tardiness Cost

Apparent Tardiness Cost (ATC) is a combination of two heuristics: the Minimum Slack (MS) and the Weighted Short Processing Time (WSPT). Firstly, the Minimum Slack (MS) heuristic aims to minimise the maximum lateness, meaning that, in a specific instance, the job to be selected should have the biggest exceeding of its deadline. [16] This variation of the Earliest Due Date (EDD) heuristic picks the minimum slack according to the computation:

$$max(d_j - p_j - t, 0)[16] \tag{2}$$

The WSPT heuristic selects the job according to the shortest processing time. However, where the Shortest Processing Time (SPT) heuristic only considers the processing times, the WSPT heuristic also considers a pre-specified weight for the jobs, which results in the computation $\frac{w_j}{p_j}$. [16] The combination of these two heuristics results in the ATC heuristic given time $t$:

$$I_j(t) = \frac{w_j}{p_j} * exp\frac{-max(d_j - p_j - t, 0)}{K\bar{p}} \quad (3)$$

Elibol et al. showed that considering the sequence-dependent setup of the ATC heuristic method, it effectively solves a single-machine scheduling problem. This Apart Tardiness Cost with Setup times (ATCS) heuristic was compared to other commonly known heuristic techniques, like the EDD and WSPT heuristics. It showed that it is significantly more effective in finding the best solution than these other heuristics. [17]

### 3.4.3 Local Search Heuristics

Next to constructive heuristics, which construct a schedule for a scheduling problem from top to bottom, there are also local search metaheuristics. These improvement metaheuristics use an existing schedule and try to improve this schedule computationally by finding better neighbours than the current schedule. [16] A neighbour of a schedule is defined as the schedule in which two adjacent jobs are switched. Essential to this is that the jobs must be done using the same machine and on the same critical path.

Examples of local search metaheuristics are Simulated Annealing (SA) and Tabu Search (TS). SA gets its name from the physical annealing process seen in metal forming, where the metal can be formed into the desired form at a high temperature and is slowly cooled down to keep that same form. Rutenbar [18] identifies the same analogy to an integrated circuit optimisation problem. These combinatorial optimisation problems can use SA to evaluate a random current solution with other solutions, thereby ending up with an optimal current situation. As with the metal analogy, the solution can change to fewer options in a high-temperature state. After which, the temperature decreases, which makes it harder to change into less optimal situations, and the probability that the optimisation yields only better solutions increases. [18] Hancerliogullari et al. [19], among others, applied the SA metaheuristic to a surgical scheduling problem. After finding an exact optimal solution employing mixed-integer programming (MIP), they found a near-optimal solution using the SA metaheuristic and compared it to the MIP solution.

Another local search metaheuristic is the Tabu Search (TS) metaheuristic; this metaheuristic differs from the standard local search metaheuristics by assigning a tabu to recently visited neighbouring solutions. This can be seen as remembering the recently visited solutions and giving them a tabu. The TS metaheuristic moves away from already visited solutions to the problem and their neighbours. [20] Therefore, the TS metaheuristic moves away from the search for local optima and searches for the global optimum. This results in a very efficient way to search for the optimal solution to a given problem.

Brandimarte [21] applied a hierarchical algorithm to a flexible job shop problem (FJSP). This metaheuristic is based on the TS metaheuristic, which assigns different memory levels during the search phase, including short, medium, and long-term memory. Brandimarte's problem deals with both routing and scheduling problems. Therefore, the different memory levels are excellent for solving this problem.

**Simulated Annealing**

Simulated Annealing was first introduced by Kirkpatrick et al. [22] in 1983. Their research tested the capabilities of simulated annealing against a travelling salesman problem, a combinatorial optimisation problem [23]. Kirkpatrick et al. showed they could identify local optima for up to N = 6000 sites, employing a decreasing factor $\alpha <= 0.95$. [22]

The Simulated Annealing metaheuristic tries to find the global optimum of an objective function by generating a random neighbour of the current system state. This system state is the initial solution of the objective function [24]. If the randomly selected neighbour provides a better (most of the time lower) solution of the objective function, the neighbour is always accepted as a new current solution. Otherwise, the neighbour is obtained according to a probability function, which depends on the current temperature and the amount of deviation of $\Delta E$ of the objective function between the neighbour and current solution [24]. The probability function for this is:

$$P(\Delta E, T) = e^{-\frac{\Delta E}{T}} \tag{4}$$

Several input parameters are required to employ simulated annealing to a scheduling problem. The first is called the decreasing factor $\alpha$. This number is used to determine the cooling schedule of the simulated annealing metaheuristic, and it should be between 0.5 and 0.99 [24]. The most used cooling schedule is:

$$T = \alpha T \text{ [24]} \tag{5}$$

Further inputs for the simulated annealing metaheuristic are the initial solution $s_0$, starting temperature $T_{max}$ and stopping criteria [24]. The initial solution $s_0$ can be any solution of the objective function. The starting temperature $T_{max}$ determines the randomness for the local search. This is caused by the use of temperature $T$ in the probability of Equation 4. At the start of the SA metaheuristic temperature $T = T_{max}$, if this temperature is very high, there is a higher probability of a worse solution being accepted. This results in a higher degree of randomness for the local search metaheuristic. A lower temperature gives a lower probability of accepting a worse solution, resulting in less randomness and moving only to local optima. This characteristic directly indicates the usefulness of the simulated annealing metaheuristic, as in the start, the metaheuristic has a broad search, narrowing the search space down when the temperature decreases and searching for the global optimum.

Lastly, the metaheuristic uses a stopping criteria. This stopping criteria can be one of the following [24]:

- A final temperature $T$, for example: $T_{stop} = 0.01$
- Stopping after several iterations, possibly without improving the best-found solution.
- Achieving a predetermined number of times a percentage of neighbours at each temperature is accepted.

#### 3.4.4 Relevance to Workshop Schedule

The heuristics, as discussed in this section, prove to be especially useful for solving scheduling problems. It discusses how heuristics can be utilised to find a nearly optimal solution for a specific scheduling problem using limited computational resources. Constructive heuristics can be used to formulate a solution schedule from scratch, considering specific scheduling constraints, like release dates, due dates and processing times. Next, improvement heuristics can be used to improve an already existing schedule. By considering a random schedule, these improvement heuristics are very efficient in finding an improved solution schedule for various scheduling problems.

### 3.5 Identification Scheduling Problem

After analysing several scheduling problems, the scheduling problem must be determined for the maintenance department, as explained in Section 2. The maintenance department consists of three mechanics with (assumed) identical processing speeds. Therefore, the scheduling problem is considered a single-stage three identical machine system. The jobs have release dates, and preemption is not allowed, so starting a job cannot be interrupted by other jobs. As this bachelor research aims to reduce overall servicing times, the objective function is set to consider the makespan of the scheduling problem. Further details of the scheduling problem will be discussed in Section 4.

### 3.6 Case Studies

During the literature study, several cases were identified with scheduling problems comparable to those of the maintenance department at Company X. The metaheuristics used in these case studies were Simulated Annealing and Tabu Search metaheuristics.

Koulamas [25] used a Hybrid Simulated Annealing metaheuristic (among others) to optimise a Parallel Machine Total Tardiness Problem. Recall from Section 3.4.2 that Tardiness is a measure of cost for scheduling jobs too late. This case study proposes two Hybrid Simulated Annealing (HSA) metaheuristics which employ the same Simulated Annealing cooling schedule but only differ from the initial solution used. Koulamas mentions that the difference in performance between HSA1 and HSA2 metaheuristics becomes more significant when the problems become more complex to solve, meaning that the initial solution significantly influences the Simulated Annealing metaheuristic. Furthermore, Koulamas compared the Simulated Annealing to another metaheuristic, combining several regular heuristics. The experiments showed that the HSA metaheuristic performs best of all compared; however, a remark was made on the significant computational requirements.

Vallada et al. compared forty different heuristics and metaheuristics on a $m$-machine flow shop problem. The objective function of this scheduling problem is to consider total tardiness. Recall from Section 3.4.3 that the Tabu Search metaheuristic tries to find a better solution by iterating on neighbouring solutions and keeping a short-term memory of recently visited neighbours. These tabu neighbours cannot be visited over a certain number of iterations. Among the forty heuristics are simple heuristics like EDD and the MS heuristics, as well as various adaptations of the Tabu Search and Simulated Annealing metaheuristics. These variations are mainly based on the way jobs are swapped. Two cases were identified here: job insertion and job interchange. Vallada et al. concluded that both the SA and TS metaheuristics performed excellently compared to the other metaheuristics, but all variants of the SA metaheuristics outperformed the TS metaheuristics. [26]

### 3.7 Summary

This section reviewed the literature on job scheduling and identified different job scheduling problems. Ways to solve job scheduling problems were also identified, mainly focusing on using heuristics, which proved to have computational advantages. The maintenance department's scheduling problem was recognised as a single-stage three, identical machine scheduling system with release dates and non-preemption.

This resulted in the research question of this stage being answered:

*What relevant theories best apply to the maintenance department to develop a schedule improvement method?*

**Relevant theories to develop a schedule optimisation method**

Several theories exist on developing a schedule optimisation method; two distinct cases can be identified: scheduling problems and knapsack problems. Scheduling problems assign jobs to resources based on time and resource constraints, whereas the knapsack problem only doesn't consider the sequence in which something is scheduled. Next, two main approaches can be identified to solve these problems: exact algorithms and heuristics. The main difference between the two is that the precise algorithm always gets the best solution, but this comes at a high cost of computational demands. On the contrary, heuristics are less demanding on a computational level and can deliver near-optimal solutions. As no due dates are used in the current process at the maintenance department, heuristics using due dates in the scheduling process cannot be used; therefore, to compare to the current (FCFS) scheduling methodology, heuristics will be used to determine a schedule based on processing time, being the Shortest Processing Time (SPT) and Longest Processing Time (LPT) heuristics. The literature review resulted in two useful metaheuristics to improve the initial solution proposed by the SPT and LPT heuristics, the Simulated Annealing (SA) and Tabu Search (TS) metaheuristics. The case studies showed that the SA metaheuristic provided a better solution than the TS metaheuristic; therefore, it was chosen to use in the solution design.

# 4 Solution Design

The current chapter will cover the formulation and design of the solution method. Section 4.1 describes the problem in detail, including objective function, parameters and constraints. Section 4.2 describes the requirements and assumptions for the solution model. In Section 4.3, the scheduling methodology is described. Section 4.4 explains the functions of the scheduling model. And lastly, Section 4.5 summarises and concludes this chapter to answer the research question:

*How can Company X develop a service schedule using the selected scheduling method?*

## 4.1 Problem Description

Company X' maintenance department is having trouble with long waiting times for customer equipment, which leads to an inefficient servicing process and schedule. Currently, the servicing process follows a First-Come-First-Serve (FCFS) principle in their schedule, but this has resulted in inefficiencies, as discussed in Section 2.2. To address this issue, a scheduling method will be developed to determine which scheduling heuristic(s) will be most effective for the maintenance department. Several parameters are required as input for this scheduling method, which will be elaborated further.

### 4.1.1 Parameters

The parameters will be used as input for the solution method; the following information will influence the servicing schedule:

- Total set of jobs: $J = \{1, ..., N\}$
- Set of mechanics operating: $M_i = \{1, 2, 3\}$
- Cost for processing service order $j$ using mechanic $i$: $p_{ij}$ with $j \in N$ and $i \in M_i$
- Release date for job $j$: $r_j$ with $r_j = \{0, 1, ..., 51\}$
- Week number $w$ for the week's schedule, with $w \in \{1, ..., 52\}$

Next to these parameters, several input settings have to be determined for the simulated annealing heuristic; these are:

- Starting temperature $T_{start}$, with $T_{start} > T_{stop}$ and $T_{start} \in \mathbf{R}^+$
- Stopping temperature $T_{stop}$, with $T_{stop} > 0$
- Decreasing factor $\alpha$, with $0.5 \leq \alpha \leq 0.99$
- Iteration length $k$, with $k \in \mathbf{N}$
- Stopping criteria $n$, with $n \in \mathbf{N}$

### 4.1.2 Completion Time

The main aim of the scheduling method is to reduce the time taken for servicing, which, in turn, means minimising the total makespan. As explained in Section 2.7, the total makespan is the duration between the start of the first operation and the end of the last operation. In Section 4.3.1, it was discussed that the scheduling method would work weekly, implying that each week has a set of operations that must be scheduled following their release date. The jobs are divided over the available mechanics, and the time taken to plan all these jobs within the given week over all available mechanics (or until no more jobs can fit in the week's schedule) is referred to as the completion time, $C^{max}_{i,w}$ for a given week $w$ and mechanic $i$. The overall completion, therefore, is the sum of the maximum completion for each week among all mechanics for weeks $W$ and mechanics $i$, ie. the sum of $C^{max}_{i,w}$ for all weeks W and mechanics i.

### 4.1.3 Objective Function

The objective function depicts the goal for which a near-optimal solution must be sought. Directly taken from Section 2.7, the solution model aims to minimise total cost, which is the total makespan for all service orders. This is formulated according to the function:

$$min \sum_{w=1}^{52} \sum_{i=1}^{3} C_{i,w}^{max} \tag{6}$$

### 4.1.4 Constraints

Several constraints were formulated in Section 2.6 to ensure the scheduling method would apply to the situation of the maintenance department:

- Working hours are limited to 8 hours (or 480 minutes) per mechanic and 40 hours per week.
- There is no room for scheduling overtime. If a job can't fit in the system, it is scheduled the next day.
- If jobs do not fit in the week their release date is, the job will receive priority in scheduling the following week.

## 4.2 Requirements Solution Model

The scheduling model should cohere with certain requirements to make it applicable to the servicing schedule. These requirements ensure that the model will be as close to the real-life situation:

- Each service order is processed once by one mechanic. This means that each job has one operation, the service order, and is serviced by a single mechanic.
- The mechanics have the same processing time, meaning each mechanic works as hard as the others.
- Priority can be given to specific service orders, based upon the reasons in Section 2.6.3.
- Overall workshop capacity is based on how many mechanics are present on a given day; either 1, 2 or 3 mechanics can be present on a given day.

### 4.2.1 Assumptions

The following assumptions were made to ensure a simplification of the problem and open the door for an efficient schedule (for further information, see Section 2.5):

- Jobs are assumed to be non-preemptive, meaning that a job cannot be interrupted.
- Release dates are all given at the start of next week; jobs can only be scheduled at the start of a week.
- Processing time for a particular job consists of standard and extra service time elements, as described in Section 2.3.3.
- Changeover time is assumed to be 5 minutes.

## 4.3 Heuristics

The solution model will employ several different heuristics, which are:

- First Come First Served (FCFS)
- Shortest Processing Time (SPT)
- Longest Processing Time (LPT)
- Simulated Annealing (SA)

### 4.3.1 Current Scheduling Method

Section 2.8 explains that the current scheduling method is based on the principle of First Come First Served (FCFS). This principle processes the jobs based on a first-in, first-out basis. It has been used to model the current situation at the servicing department, as it aligns with the primary way of its operations. The schedule is made on a week-by-week basis, which is also how the company operates, and this is also considered in the solution model. The FCFS principle sorts the incoming jobs in an ascending order of order date to generate the schedule. As there is no record of this scheduling method, no real-life comparison can be made for the FCFS principle. Therefore, the FCFS principle is modelled to resemble the current scheduling method.

As Section 2.8 mentions, the current scheduling method is based upon some assumptions. First, each job will have a release date assigned, which denotes the week the jobs are made available for scheduling. Furthermore, changeover time is considered to be 5 minutes, meaning that the setup time for a job is always 5 minutes.

### 4.3.2 Shortest Processing Time

The Shortest Processing Time heuristic uses the same scheduling model as the FCFS principle. The only key difference is that the SPT heuristic sorts the to-be scheduled jobs on an ascending order of processing time instead of the order date from the FCFS principle. This will mean that the shortest jobs will be dealt with first, after which the processing time will increase gradually until all jobs are completed. According to Holthaus, the SPT heuristic is an excellent dispatch rule to minimise job tardiness [27].

### 4.3.3 Longest Processing Time

The Longest Processing Time heuristic is precisely the same as the SPT heuristic; the only difference is that the jobs are sorted in descending order of processing time, prioritising jobs with a long expected processing time.

### 4.3.4 Simulated Annealing

Simulated Annealing (SA) is a unique heuristic that is different from other heuristics. The main goal of this heuristic is to improve an existing schedule. This form of improvement heuristics takes any schedule as its initial solution and tries to improve it. Improvement is carried out by finding the neighbours of the schedule that differ in one aspect, i.e., two jobs are interchanged. The SA heuristic selects a random neighbour of the current solution to achieve this objective and examines whether it improves the objective function. When a better solution is found, that solution becomes the current solution, and the examination process starts again. To reduce the randomness of selecting a random neighbour, the SA heuristic uses a decreasing factor $\alpha$. This factor (in combination with $T_{start}$ and $T_{stop}$) makes the cooling schedule of the SA metaheuristic. This cooling schedule gradually reduces the temperature from a high value ($T_{start}$) to a low value ($T_{stop}$), which reduces the impact of randomly selecting, which helps to widen the search area and find an optimal solution [28].

## 4.4 Scheduling Model

The scheduling model will use the discussed heuristics in a formulated schedule using constructive heuristics and further improve the heuristics using simulated annealing. This section will discuss all microflows and the input used to generate a schedule.

### 4.4.1 Scheduling Input

The input for the scheduling model directly comes from Company X' historical job data. For the solution model, data from the year 2023 has been analysed, as this best reflects the situation of Company X. Service order data was taken and analysed according to the process described in Section 2. Service orders are input for the scheduling model from this data, and information about each job is gathered as follows.

The order number is required to schedule a job, and processing times and release dates must be assigned. The order number directly comes from the list of historical service order data. Next, the processing time is assigned, and the service order is based upon the standard maintenance orders and/or the categorised analysis of the service times, as discussed in Section 2.3.3. Either one of the two components denotes the time, or both represent the service time. The service time is taken in minutes and is assigned to the order number. Furthermore, a release date is assigned based on the order date. As discussed in Section 2.5, it is assumed that all orders falling within a week can be scheduled at the start of that week. This release date is denoted by an integer value between 1 and 52.

### 4.4.2 Run Scheduling

First, the microflow *RunScheduling* is called to start the scheduling process. This microflow initiates (and clears) lists *MechanicJobs* and *PerformanceList*; these lists are used to store the scheduled jobs and KPIs, respectively. Furthermore, it calls the sub-microflow *GenerateMechanicSchedule*, which generates a schedule for each day a mechanic is available in the workshop whilst considering public holidays and days on which on-site maintenance is performed (see Section 2.4.2 for further explanation).

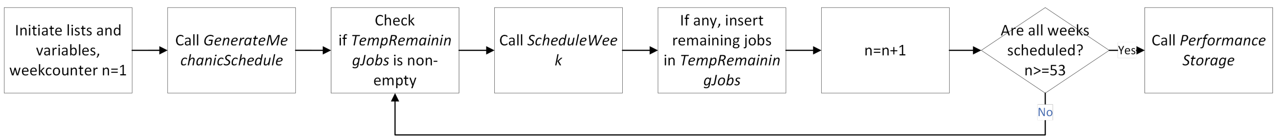


Figure 15: *RunScheduling* microflow

Next, the microflow calls the sub-microflow *FCFS ScheduleWeek*, which generates a schedule for the week based on the FCFS principle. This microflow returns if any jobs are left from the past week; if this is the case, these jobs are prioritised to schedule the following week. Lastly, after schedules are generated for each week, the microflow calls the sub-microflow *PerformanceStorage*, which stores the KPIs for all weeks and mechanics for further evaluation in the list *PerformanceList*.

### 4.4.3 Schedule Week Microflow

The *ScheduleWeek* microflow is called by the *RunScheduling* microflow, which delivers input parameters for the current microflow. These parameters for setup time, current week number and remaining jobs from last week ensure that the week's schedule is effectively generated.

The microflow sorts all jobs for the current week on order date to ensure the FCFS principle; this also ensures that jobs from last week (if any) are given priority before the jobs from the current week. These jobs are sorted on processing times if the SPT or LPT is used, either in ascending or descending order. After this, the schedule generation is performed. First, it finds all mechanic schedules generated by the *GenerateMechanicSchedule* sub-microflow. The microflow picks the first job in the *JobsList*, sees the first available mechanic, and schedules the job for that mechanic. Furthermore, a place is made for the job in the *MechanicJobs* list in which details are stored for each job assigned to a mechanic, including order number, planned duration, assigned mechanic, week number and processing date. After this, the mechanic's availability (remaining processing time for the day) is updated, and the job is removed from the *JobsList*.

If the case a job cannot be found in any of the mechanics' schedules for the current day, a new day is found. If this day is the same week as the present day, microflow keeps iterating the jobs. If this day is in the next week, the microflow is stopped, and all remaining jobs are stored in a temporary list to be added to the *JobsList* of next week. If all jobs in the *JobsList* are scheduled, the microflow is ended and returned to the *RunScheduling* microflow to start with the following week.

### 4.4.4 Generate Mechanic Schedule

The *GenerateMechanicSchedule* microflow generates a new schedule for each mechanic. This schedule is used to assign jobs to an available mechanic. This schedule is generated by iterating over the days of the year and checking the availability of each mechanic. If the day is NOT a weekday, no schedule can be made; if the day is a public holiday, no schedule can be made. Furthermore, it checks if the day has either one, two or three mechanics, depending on the performance of on-site maintenance. The generated mechanic schedules are stored in the list *ScheduleList*, used in the *ScheduleWeek* microflow.

### 4.4.5 Simulated Annealing Microflow

Simulated Annealing is a technique that involves swapping operations to find a better and eventually near-optimal solution. To find this solution, the SA heuristic is initialised, the temperature settings $T_{start}$ and $T_{stop}$, iteration length $k$ and the decreasing factor $\alpha$ are set. The initial solution $s_0$ is determined. If the current temperature $T$ is higher than $T_{stop}$ the search is started. The heuristic determines for a preset number of iteration $k$ the number of times it iterates with the current settings. A randomly selected neighbour is determined and evaluated over its completion time $C_{max}$. If the completion time is lower, the neighbour is always accepted. If the completion time is higher, the neighbour can be accepted according to the probability of Equation 4. In all cases after the evaluation of the neighbouring solution, a counter $i$ is incremented by one. This counter determines if the iteration length $k$ is reached. If so, the temperature is decreased according to Equation 5. If the iteration length $k$ is not yet reached, another iteration is carried out. As can be seen in Figure 17, the SA heuristic is stopped as soon as the temperature $T$ reaches a value lower or equal to the temperature of $T_{stop}$. Next to the cooling schedule formulated by the value of $T_{start}$, $T_{stop}$ and iteration length $k$, a stopping criteria is considered. This stopping criterion accounts for the number of calculations made for no improvement. If this preset number of calculations without improvement is reached, the SA heuristic is terminated. The current solution, then, is the result of the SA heuristic.

In the scheduling model, all operations are based on weekly. First, the microflow *SimulatedAnnealing* initialises the settings for the SA heuristic; it always uses the last entered settings put into the *SA Settings* entity. Next, this microflow is called the *SimulatedAnnealingWeek* microflow, which retrieves the list of scheduled jobs for the given week. This *InitialSolution* is the input for the SA heuristic. The best solution is searched in the *DetermineBestSolution* microflow, which inputs the initial solution and follows the logic in Figure 17 to find the best solution.

To do this, two random numbers are generated to find a random neighbour solution of the initial solution. These random numbers are chosen from the amount of jobs in the week's schedule. These random numbers denote the positions of two randomly selected jobs. For instance, if the schedule for the week has 36 jobs, the random numbers will be picked from 1 to 36. If the two random numbers are the same, one random number is regenerated. These random numbers determine the scheduled jobs that will be swapped to define a neighbour of the initial solution. The jobs are swapped

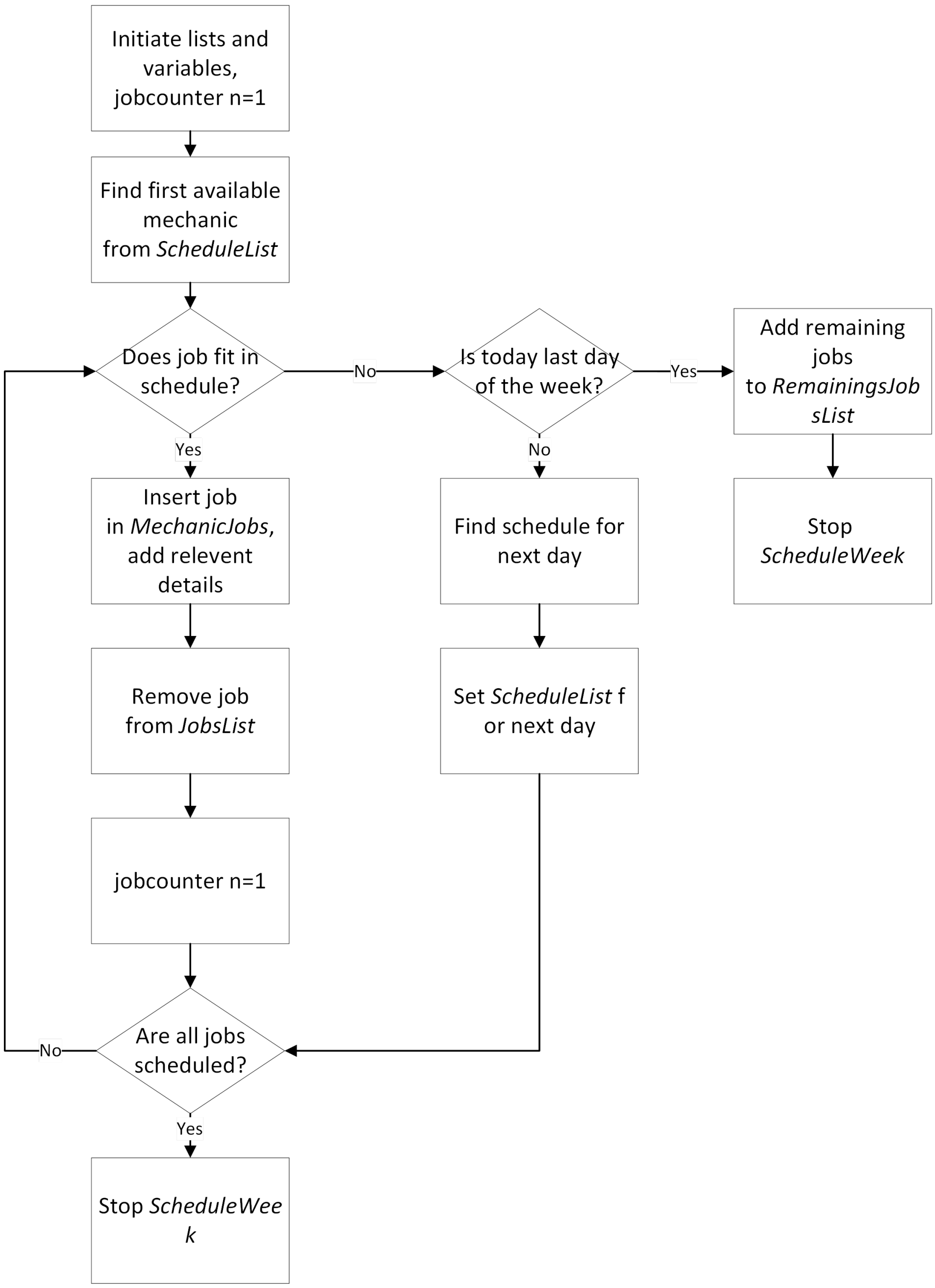


Figure 16: *ScheduleWeek* microflow

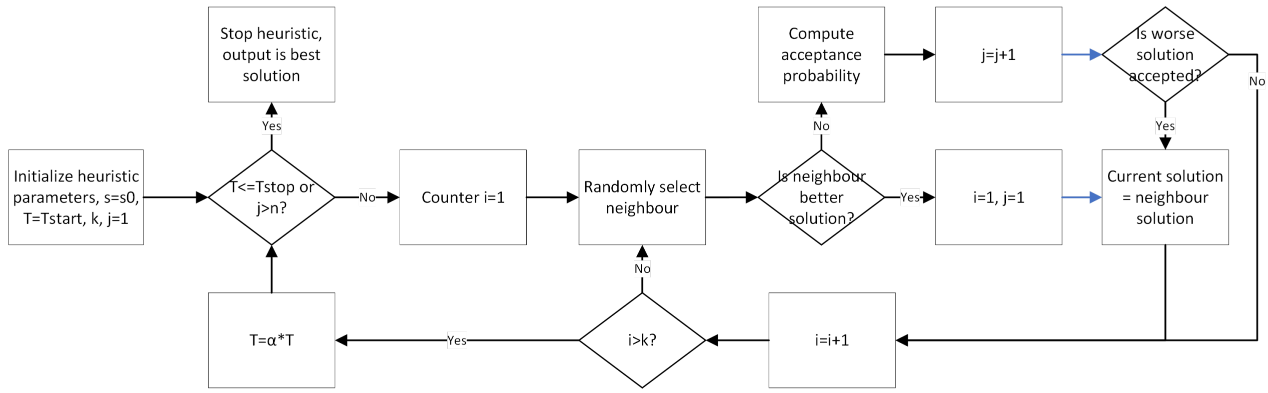


Figure 17: Simulated Annealing heuristic flowchart

by interchanging the date and mechanic assignments, and the associated mechanic schedules are exchanged. To check if this possible neighbour is valid, it is determined if one of the mechanics is not over their capacity. If a mechanic is over their capacity, the swapping is stopped, and another cycle of neighbour calculation is started until a valid neighbour is found. The week's schedule is retrieved for the neighbouring solution, which is needed to determine the completion time, as this is the variable on which the solution is evaluated.

If the swapping of operations does result in a valid neighbour, the completion time is calculated for the given week. The completion of the neighbour is then compared to the current solution; if it results in a better (lower) completion time, the neighbour is accepted as the current solution. The acceptance probability is computed if it results in a worse (higher) completion time. Using the rand() function in Mendix, a random decimal number between 0 and 1 is generated; the worse solution is accepted if the random number is higher than the acceptance probability. If that is the case, the generated solution is accepted as the current solution. A counter variable $i$ is increased by one, and if it is not higher than the iteration length $k$, another neighbour is sought, and the steps mentioned above are repeated once again. If the value $i$ exceeds the pre-specified value of $k$, the temperature $T$ is multiplied by the decreasing factor $\alpha$. If the temperature $T$ has reached a value lower or equal to $T_{stop}$, the microflow will be terminated, and the current solution is the best-found solution for the schedule of the given week. This annealing process can be terminated prematurely by the stopping criteria, which terminates if the annealing process hasn't produced an improved solution during the last $n$ computations. This means that after $n$ computations without improvement, the annealing process will be stopped. This process is performed each week according to the schedule.

### 4.4.6 Performance Storage & Weekly Statistics

Key Performance Indicators (KPIs) are used to gather information about the schedule. The information necessary for calculating these KPIs is integrated into the solution model using the *PerformanceStorage microflow*. Once all jobs have been scheduled, the *RunScheduling* microflow calls the *PerformanceStorage* microflow to collect information about the performance of the schedule. This information primarily includes the capacity remaining in each mechanic's schedule, which is used to determine their idle time. Using this remaining capacity, the overall working time per mechanic can be calculated, which will be used to determine the KPIs.

The KPIs are calculated in the *ComputeKPIs* microflow, which computes the KPIs discussed in Section 2.7. The total makespan is calculated by determining the completion time for each week and adding up all these weeks' aggregate sums. The average daily idle time per mechanic is then calculated to determine how long a mechanic has been idle. For this purpose, a weekly average per mechanic is taken to provide a better overview of the maintenance department's efficiency. Additionally, a total processing time is calculated each week to measure productivity. These KPIs are stored on the *Weekly Statistics* page, where all weeks can be compared.

## 4.5 Summary

This chapter has formulated a solution model for the maintenance department. It first described the scheduling problem, including problem description, completion time description, objective function and constraints. Next, the requirements for the solution model are elaborated on, including vital details like processing speeds and job characteristic assumptions. After that, the scheduling methodologies used in the solution model are described. Lastly, the functions of the scheduling model are described, including details like mechanic schedule generation, scheduling microflows and KPI registration.

Furthermore, this chapter has answered the research question:

*How can Company X develop a service schedule using the selected scheduling method?*

**Developing a service schedule**

The service schedule is developed through the Mendix application designed in this chapter. The application uses several constructive heuristics in combination with the SA heuristics to determine a service schedule for the maintenance department. Section 4.4.5 describes how the SA metaheuristic determines a better solution for its initial solution, generated by one of the three constructive heuristics.

# 5 Evaluation & Implementation

This chapter will evaluate the solution model designed in the previous chapter and formulate an implementation plan. Section 5.1 discusses the experimental design, and the experiments are carried out in Section 5.2 to determine the input settings for the SA metaheuristic. Section 5.3 shows the workings of the SA metaheuristic. The input settings will be applied in Section 5.4 to evaluate the scheduling methods. Then, Section 5.5 formulates an implementation plan for the solution model at the maintenance department. Lastly, Section 5.6 summarises this chapter and reports the main conclusion of the evaluation to answer the research question:

*What recommendations and conclusions can be made after implementing the servicing methodology and schedule?*

## 5.1 Experimental Design

Multiple experiments were conducted to evaluate the effectiveness of the solution model. These experiments are designed to test the various heuristic approaches used in the model. This section details the experiments and the dataset used to test the solution model.

The first set of experiments will focus on the input settings for the SA heuristic. Several experiments will be carried out to test these input settings. Section 3.4.3 provided the necessary information regarding the parameters needed for the SA metaheuristic. The SA input settings experiment will be discussed in Section 5.2.

After the input settings are determined, the heuristics will be evaluated in Section 5.4. First, the constructive heuristics, specifically the FCFS, SPT, and LPT heuristics, will be compared to one another. The aim is to compare these heuristics and determine which is the most effective in the solution model. Next, an evaluation is carried out to compare the constructive heuristics combined with the SA metaheuristic to the current (FCFS) scheduling approach.

This section further discusses the experiments, KPIs, and dataset used to conduct the experiments.

### 5.1.1 Dataset

The experiments used a dataset comprising service data from the previous year (2023). The original dataset contains all service lines from last year's service orders. As a result of Section 2, the dataset was created, which now can be imported into the solution model. The processing times for the operations that required scheduling were determined based on the service times in this dataset. A service job's service time was divided into standard and extra service time, explained in more detail in Section 2.3.3. It should be noted that these service times do not include the switching time between jobs, which has been added to the service time. The dataset contains 1536 service lines, meaning that 1536 service orders must be scheduled for testing in the year 2023, which averages 76.8 service orders per week.

Before starting the experiments, the Microsoft Excel file is imported into the solution model using the Mendix Excel importer and Mx Model Reflection module from the Mendix marketplace. These modules are built-in utilities developed by Mendix. The Mendix Excel importer uses a manually created template to import the Excel file into the Mendix application. The Mx Model Reflection was needed for the Excel importer to function correctly. This module makes information about the application's domain model accessible from the application.

### 5.1.2 Results & KPIs

Gaining insight from the output of the solution model is vital for evaluating the solution. Next to the KPIs mentioned below, some weekly results are shown. These results will discuss weekly statistics regarding the used heuristics, which visually represent these weekly statistics. The Key Performance Indicators are used to measure the performance of different heuristics and compare the heuristics to each other. The KPIs are as follows:

- Total makespan; The total time to process all jobs in the schedule (in minutes). Calculated by adding up the completion time of all weeks together. This value is important as it shows the overall efficiency of the scheduling model; a lower makespan is preferable.
- Total idle time; the time mechanics aren't performing any tasks (in minutes). Only days are considered for which at least one job is scheduled.
- (Weekly) average daily idle time; average idle time a mechanic spends daily while not performing any tasks. The average daily idle time is considered per mechanic. By determining the weekly average, insight can be drawn into the performance in different weeks.
- Computation time; the computation time taken to run the scheduling heuristics (in seconds)

### 5.1.3 Experiment Validation

This section considers the validation of the experiments, covering the computational setting and randomness of the heuristics.

**Computational Setting**
The experiments will be carried out on the same Windows computer, having 16 GB of RAM and an Intel Core i7-8750H processor @ 2.20 GHz. All experiments are carried out in Mendix 10.7.0.

**Randomness**
The SA heuristic uses probability to determine the neighbouring solution and the acceptance of a worse solution. Therefore, all experiments will be performed ten times to account for this randomness.

## 5.2 SA Settings

This section will discuss the settings used for the SA heuristic and the tests conducted to examine the best settings. As mentioned in Section 4.4.5, the SA heuristic requires the following input settings:

- Starting temperature $T_{start}$
- Stopping temperature $T_{stop}$
- Decreasing factor $\alpha$
- Iteration length $k$
- Stopping criteria

Several experiments were conducted to find an optimal set of capable input settings for the SA heuristic. Each experiment will evaluate different input settings; the experiments will be reviewed on the KPIs *total makespan* and *computation time*, as discussed in Section 2.7. The total makespan KPI was chosen to determine the overall effectiveness of the heuristic, as it directly represents the objective function. The computation time was selected to consider the computational efficiency, as it indicates the impact of the input settings on the scheduling model. When testing a particular input setting, arbitrary values will be set for the other input settings to evaluate the current input setting.

Table 4: KPIs starting temperature

| Temperature | Total makespan (min) | Computation time (sec) |
|---|---|---|
| 10 | 38390.80 | 84.21 |
| 25 | 38335.80 | 86.07 |
| 50 | 38322.30 | 90.46 |
| 100 | 38394.30 | 92.57 |
| 250 | 38583.80 | 96.80 |
| 500 | 38455.10 | 102.12 |
| 1000 | 38571.70 | 106.35 |
| 2000 | 38539.80 | 108.95 |
| 2500 | 38608.70 | 107.79 |

### 5.2.1 Starting Temperature

The starting temperature $T_{start}$ can be any value it only must be higher than $T_{stop}$. For this experiment, the arbitrary values for the other input settings are $T_{stop} = 0.01$, $\alpha = 0.80$ and $k = 1$. Due to the nature of the acceptance probability function (Equation 4), a higher value of $T_{start}$ results in a broader search space in the cooling schedule. The values for experiment for the starting temperature $T_{start} \in \{10; 25; 50; 100; 250; 500; 1000; 2000; 2500\}$. This results in the KPIs' values in Table 4.

The results in Table 4 show that the total makespan does not change a significant amount when adjusting the value of $T_{start}$, where the percentual difference between the highest makespan (38608.70 minutes) and the lowest makespan (38322.30 minutes) over ten repetitions is only 0.747%. This result shows that the starting temperature doesn't significantly influence the total completion time for the scheduling model. However, the starting temperature determines how broad the search area of the simulated annealing heuristic is. Therefore, it is believed that a starting temperature that is high enough must be set to ensure that the SA heuristic can move away from a local optimum and move towards the global optimum of the scheduling model. Therefore, it is decided that a starting temperature $T_{start} = 500$ is chosen as this will ensure a broad search space, whilst also considering the computation time in seconds.

### 5.2.2 Stopping Temperature

The stopping temperature $T_{stop}$ can be any positive value, and it must be lower than $T_{start}$; in retrospect to the starting temperature, the stopping temperature has importance to the length of the improvement on a heuristic. To evaluate the value of $T_{stop}$ values for the other inputs settings were set at $T_{start} = 2000$, $k = 1$ and $\alpha = 0.80$. The values for the experiment of the stopping temperature were sat at $T_{stop} \in \{0.01; 0.02; 0.05; 0.10; 0.25; 0.50; 1.00; 1.50; 2.00\}$

Table 5 shows the results for the evaluation of the stopping temperature for the SA heuristic. The total makespan significantly improves when using a gradually lower value for $T_{stop}$. The lowest possible value that can be input in the scheduling model has two decimals; therefore, the lowest possible value for $T_{stop}$ is 0.01. This also results in the lowest value for the total makespan; thus, the $T_{stop}$ is set to 0.01.

Table 5: KPIs stopping temperature

| Temperature | Total makespan (min) | Computation time (sec) |
|---|---|---|
| 0.01 | 385398 | 108.954 |
| 0.02 | 386588 | 106.499 |
| 0.05 | 387363 | 97.818 |
| 0.10 | 389801 | 95.593 |
| 0.25 | 391578 | 93.878 |
| 0.50 | 393773 | 91.511 |
| 1.00 | 396736 | 84.999 |
| 1.50 | 398738 | 85.172 |
| 2.00 | 400259 | 85.68 |

### 5.2.3 Decreasing Factor

Next, the decreasing factor is evaluated, determining the SA heuristic's cooling schedule. Equation 5 shows that the decreasing factor determines the temperature level. Therefore, a lower value $\alpha$ results in shorter cooling schedules and less room for improving the schedule. A higher value $\alpha$ gives more time in the cooling schedule to determine the best value of the schedule. For the experiment the following input settings were used $T_{start} = 2000$, $T_{stop} = 0.01$, $k = 1$ and $\alpha \in \{0.50; 0.60; 0.70; 0.80; 0.90; 0.92; 0.95; 0.97; 0.99\}$

Table 6: KPIs decreasing factor

| Decreasing factor | Total makespan (min) | Computation time (sec) |
|---|---|---|
| 0.5 | 397235 | 79.571 |
| 0.6 | 394649 | 72.653 |
| 0.7 | 391081 | 81.865 |
| 0.8 | 385946 | 99.627 |
| 0.9 | 378405 | 155.313 |
| 0.93 | 375961 | 204.619 |
| 0.95 | 374034 | 284.156 |
| 0.97 | 371656 | 478.982 |
| 0.99 | 369475 | 1337.754 |

Table 6 shows the results for the KPIs from the experiment regarding the decreasing factor. The table shows that a higher value of $\alpha$ reduces the total makespan and increases the computation time. The computation time increases as a higher $\alpha$ causes a more extensive set of computations to perform the SA heuristic. Logically, $\alpha = 0.99$ results in the best solution; however, the computation time is nearly 2.8 times longer than the computation time for $\alpha = 0.97$. Considering the model's computational efficiency, $\alpha$ is set to 0.97.

### 5.2.4 Iteration Length

The iteration length $k$ can be any natural number, and the value of $k$ determines how often the SA heuristic iterates for a given temperature value. However, the values of $k$ heavily impact the computational efficiency of the heuristics, as it directly influences the number of calculations. To determine the best value of $k$, the following input settings were used: $T_{start} = 2000$, $T)stop = 0.01$ and $\alpha = 0.80$. The values considered for the iteration length $k \in \{1; 2; 5; 10; 20; 30; 50\}$.

Table 7: KPIs iteration length

| Iteration length | Total makespan (min) | Computation time (sec) |
|---|---|---|
| 1 | 383908 | 84.206 |
| 5 | 375642 | 214.867 |
| 10 | 372360 | 389.575 |
| 20 | 370253 | 667.124 |
| 30 | 370558 | 928.707 |
| 50 | 369683 | 1275.616 |

Table 7 shows that the total makespan decreases when the value of $k$ increases. Therefore, it would sound logical to pick a value as high as possible for the iteration length. However, increasing the iteration length also demands many more calculations, increasing the computation time. Table 7 shows that increasing the value of k from 20 to 50 only slightly improves the total makespan but drastically increases the computation time by 91.2 %.

### 5.2.5 Stopping criteria

The stopping criteria, as discussed in Section 4.4.5, considers the number of computations without having getting an improvement. The number of computations $n$ can be any natural number and strongly correlates with the number of computations of the overall SA heuristic. To determine the best value of $n$, the following input settings were used: $T_{start} = 2000$, $T)stop = 0.01$, $\alpha = 0.80$ and $k = 10$. The values considered for the stopping criteria $n \in \{10, 25, 50, 100, 200, 250, 300, 400, 500\}$.

Table 8: Stopping criteria

| Stopping criteria (n) | Total makespan (min) | Computation time (sec) |
|---|---|---|
| 10 | 417966 | 81.47 |
| 25 | 398961 | 165.75 |
| 50 | 388323 | 242.12 |
| 100 | 384979 | 313.45 |
| 200 | 383300 | 323.83 |
| 250 | 383288 | 337.16 |
| 300 | 383801 | 347.39 |
| 400 | 383542 | 341.96 |
| 500 | 383308 | 348.51 |

Table 8 indicates that the total makespan decreases as $n$ increases, while the computation time increases with the increase in the value of $n$. The KPIs show a significant change in the initial increments of the stopping criteria, after which they tend to stagnate for higher values of $n$. The best total makespan can be achieved when the stopping criteria for $n$ is set to 250, beyond which the total makespan slightly increases for higher values of $n$. The stopping criteria gradually increase, and so does the computation time with a higher value of $n$.

### 5.2.6 Input Settings

As the input settings have been evaluated, the settings can be used in the experiments to evaluate the heuristics used in the scheduling model. The input settings are:

- Starting temperature $T_{start} = 500$
- Stopping temperature $T_{stop} = 0.01$
- Decreasing factor $\alpha = 0.97$
- Iteration length $k = 20$
- Stopping criteria $n = 250$

## 5.3 Proof of Concept

This subsection is meant to give a proof of concept for the simulated annealing heuristic. Section 3.4.3 describes how the simulated annealing heuristic tries to find a globally best solution by evaluating the whole search space. This is done by accepting worse neighbours according to the probability function of Equation 4. So, over time, the completion time of a schedule should fluctuate by going up and down, but ultimately, it should end in the global best of the objective function. Figure 18 shows this phenomenon using the SA heuristic for the first week of scheduling.

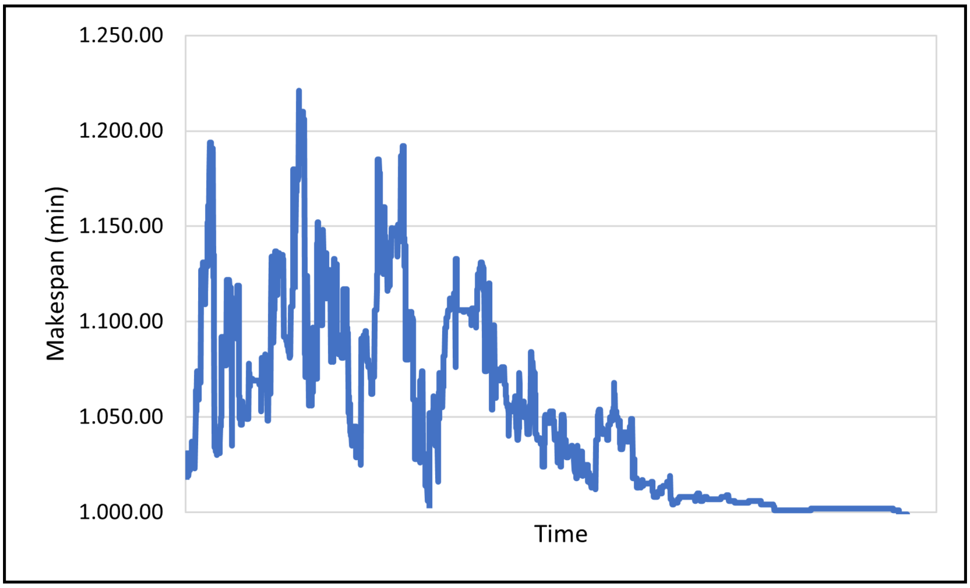


Figure 18: Evolution of makespan over time

## 5.4 Evaluation

The evaluations will be carried out in this section, and the results and KPIs will be compared. The heuristics used for the experiments are:

- FCFS: First Come First Served heuristic

- SPT: Shortest Processing Time heuristic
- LPT: Longest Processing Time heuristic
- FCFS + SA: First Come First Served heuristic in combination with the Simulated Annealing heuristic
- SPT + SA: Shortest Processing Time heuristic in combination with the Simulated Annealing heuristic
- LPT + SA: Longest Processing Time heuristic in combination with the Simulated Annealing heuristic

### 5.4.1 Constructive Heuristic Evaluation

Firstly, the three constructive heuristics used to generate a weekly schedule for the provided dataset for the year 2023 will be evaluated. Table 9 shows the results of the three KPIs concerning these heuristics; all KPIs are in minutes. As indicated, each heuristic's total processing time KPI is the same. This is instead a control KPI to see if all heuristics processed the same amount of jobs.

The following KPI, daily average idle time per mechanic, shows the time a mechanic is not working on any service orders per day. As Table 9 shows, the FCFS and SPT heuristics have little difference in this KPI with a percentual difference of 1.14%. The LPT heuristic significantly improves compared to the FCFS and SPT heuristics, with a percentual difference of -10.47% and -11.48%, respectively.

Lastly, the total makespan directly related to the solution model's objective function shows in line with the idle time KPI that the FCFS and SPT heuristic are closely related, showing a 2.72% better solution for the FCFS heuristics in comparison to the SPT heuristic. The LPT heuristic also offers better results regarding the makespan KPI, showing a -1.20% and -3.81% improvement compared with the FCFS and SPT heuristics, respectively.

Table 9: Constructive heuristics results

| | Total processing time | Daily average idle time per mechanic | Total makespan |
|---|---|---|---|
| FCFS | 90398 | 140.05 | 40247 |
| SPT | 90398 | 141.64 | 41340 |
| LPT | 90398 | 125.39 | 39766 |
| Δ% SPT-FCFS | - | 1.14 % | 2.72 % |
| Δ% LPT-FCFS | - | -10.47 % | -1.20 % |
| Δ% LPT-SPT | - | -11.48 % | -3.81 % |

Furthermore, Appendices A, B, and Figure 19 give visual representations of the results of the FCFS, SPT and LPT heuristics, respectively. Comparing these heuristics using Figure 21 shows that the makespan of most weeks is quite similar to each of the heuristics, where the SPt heuristic shows slightly higher completion times and the LPT heuristics shows a little slower completion time, which results are shown in Table 9. All three heuristics show that weeks 50, 51, and 52 have a low makespan, which may be caused by the few jobs entering the system.

Another comparison can be drawn from the idle time ratio to total processing time. The idle time ratio is the time the mechanics do not spend working on jobs in the scheduling system. Days where no jobs are scheduled aren't considered to be idle time. Figure 19 shows these ratios for the FCFS, SPT and LPT heuristics. The figure shows that the FCFS has the highest idle time ratio of 29.5%. Next, the SPT and LPT heuristics have an idle time ratio of 28.7% and 26.7%, which is a 0.8% and 2.8% improvement over the FCFS heuristic. The percentage of processing time shows the ratio of time for which jobs are processed in the scheduling model. This means that the LPT heuristic has the best mechanic utilisation with 73.3%.

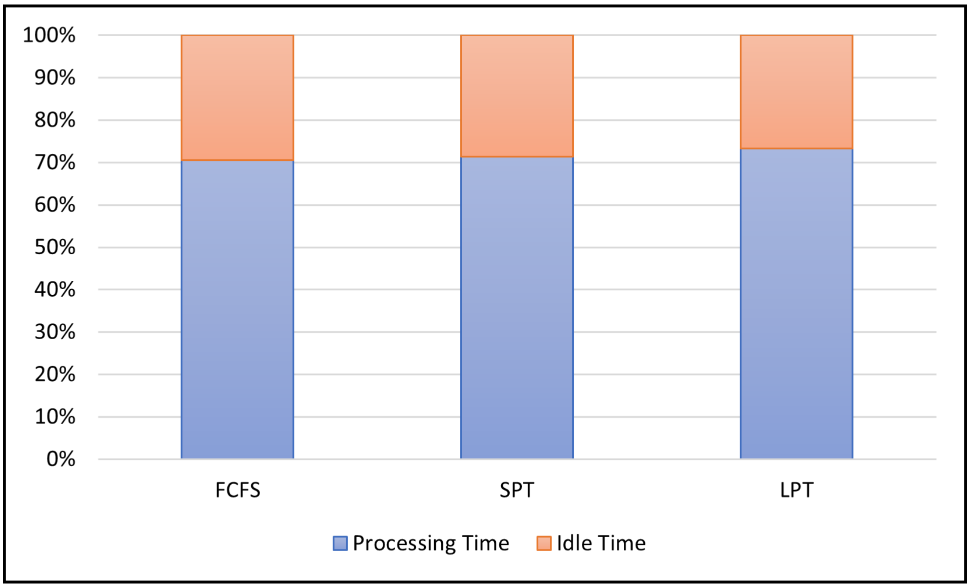


Figure 19: Idle & Processing Time Comparison

Lastly, the daily average idle time per mechanic per week is shown in Figure 22. The graph shows the average daily idle time per mechanic for each of the three constructive heuristics. It shows that most weeks have the same idle time for each heuristic. However, the LPT heuristic shows decreased idle times for the year's second half. The SPT heuristic shows higher and lower weekly idle time averages than the FCFS heuristic. This results in almost the same overall average idle time, as shown in Table 9. The LPT, however, only has a higher average idle time in week 12; in all other weeks, a difference occurs with the FCFS heuristic. These occurrences are mainly in the latter half of the year, where weeks 28, 31 and 32 have an average idle time below 15 minutes a day per mechanic. This shows that the efficiency of the mechanics of the LPT heuristic is significantly better than the other heuristics. What all heuristics have in common is an increased idle time for weeks 50, 51 and 52; this correlates with the decreased makespan in the same weeks. Therefore, it can be said that during these weeks, few jobs are entering the system.

#### 5.4.2 Improvement Heuristic Evaluation

The improvement of the heuristic experiment will evaluate the performance of the simulated annealing heuristic in combination with each of the constructive heuristics (i.e. FCFS, SPT and LPT). Each heuristic combination will be evaluated on the KPIs mentioned in Section 2.7. To assess these heuristics, one of the constructive heuristics is run, after which the SA heuristic is used; the results values for the KPIs will follow as results of the combination of the two heuristics. The settings used for the simulated annealing improvement heuristic are (as discussed in Section 5.2.6):

- $T_{start} = 500$
- $T_{stop} = 0.01$
- $k = 20$
- $\alpha = 0.97$
- $n = 250$

The experiment results of combining the simulated annealing heuristic with each constructive heuristic are shown in Table 10. The three improvement heuristics are evaluated based on the current FCFS scheduling method and the best performing LPT heuristic from the first experiment What becomes immediately apparent is that all three new ways of scheduling service orders are more efficient than the current FCFS way of scheduling. The FCFS + SA heuristic provides an 8.33% total completion time than the FCFS heuristic. Following that shows the SPT + SA heuristic, which gives a 5.39% better solution than the FCFS heuristic. And lastly, the LPT + SA heuristic gives a 5.89% improvement over the FCFS heuristic. Compared to the LPT heuristic, the three improvement heuristics performed a bit worse. Having a 7.22% improvement for the FCFS + SA heuristic over the LPT heuristic. Next, the SPT + SA and LPT + SA heuristics showed a 4.24 % and 4.75 % improvement over the LPT heuristic.

Table 10: Improvement heuristic experiment

| | **Total processing time** | **Idle time per mechanic** | **Total makespan** | **Computation time (sec)** |
|---|---|---|---|---|
| **FCFS** | 90398 | 140.05 | 40247 | 5.15 |
| **LPT** | 90398 | 125.39 | 39766 | 5.48 |
| **FCFS+SA** | 90398 | 140.55 | 36896 | 203.22 |
| **SPT+SA** | 90398 | 141.96 | 38078 | 214.87 |
| **LPT+SA** | 90398 | 125.42 | 37878 | 201.97 |
| **% (FCFS+SA)-FCFS** | 0 | 0.36% | -8.33% | 3846% |
| **% (SPT+SA)-FCFS** | 0 | 1.36% | -5.39% | 4072% |
| **% (LPT+SA)-FCFS** | 0 | -10.45% | -5.89% | 3822% |
| **% (FCFS+SA)-LPT** | 0 | 12.09% | -7.22% | 3608% |
| **% (SPT+SA)-LPT** | 0 | 13.22% | -4.24% | 3821% |
| **% (LPT+SA)-LPT** | 0 | 0.03% | -4.75% | 3586% |

Next, the average idle times can be compared for each heuristic. In line with the average idle times of solely the SPT and LPT constructive heuristics to the FCFS heuristic, the SPT + SA and LPT + SA heuristics show similar results regarding the average daily idle time per mechanic, with the SPT + SA heuristic having 1.36% increase in average idle time. The LPT + SA heuristic has a 10.45% improvement over the FCFS heuristic. Lastly, the FCFS + SA heuristic gave a minor decrease of 0.36% to the idle time per mechanic. The improvement heuristics didn't show any improvement in the average idle times for each heuristic, having a respective 12.09%, 13.22% and 0.03% increase of the idle time KPI for the FCFS + SA, SPT + SA and LPT + SA heuristics compared to the LPT heuristic.

Lastly, the computation times can be compared. These clearly show that the simulated annealing takes way more time, giving the heuristics that use this method over 35 times more computation time than the FCFS heuristic. This aligns with expectations as simulated annealing requires significantly more computations for a better solution.

Appendices C, D and Figure 20 give visual representations of the results of the experiment regarding the simulated annealing heuristic. Figures 23, 24 and 25 compare each of the constructive heuristics with the combination of constructive heuristics and Sa heuristic. For most weeks, each picture shows the SA heuristic results in a lower makespan (in minutes). Only the LPT heuristic occasionally gives a higher makespan when combined with the SA heuristic. The FCFS and SPT heuristics only have a higher makespan in week 29 when combined with the SA heuristics. All other weeks give a lower makespan. Figure 26 shows each of the combinations with the SA heuristic compared to each other. What becomes apparent is that in most of the weeks, the makespan is quite similar for each heuristic. This could mean no further improvement can be made for those weeks, regardless of the heuristic.

Figure 27 compares the average daily idle time per mechanic per week. The graph looks very similar to the graph in Figure 22 as most weeks do not differ in idle time. This also becomes apparent when comparing Figure 20 to Figure 19. The ratios of idle time are analysed using one decimal, and the same idle time ratio is shown.

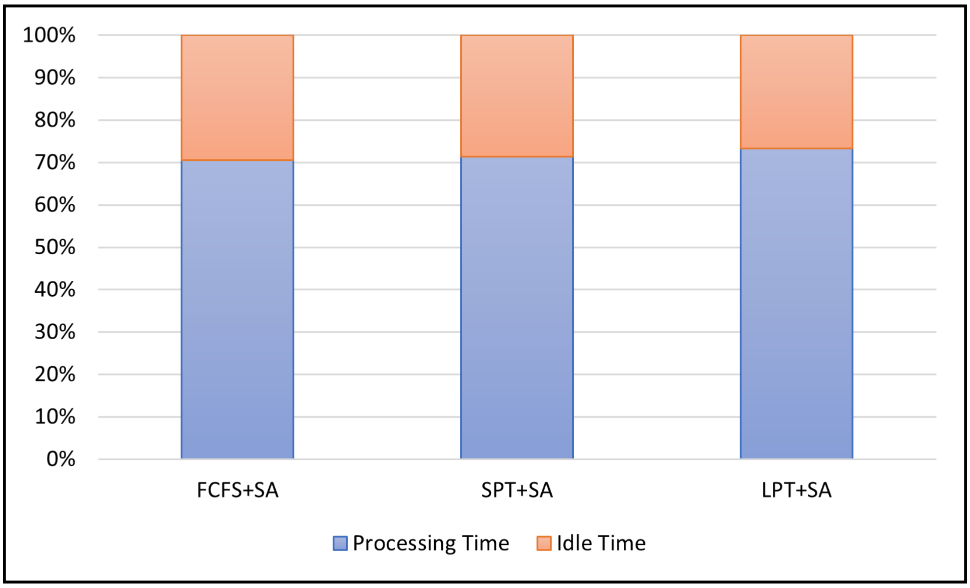


Figure 20: Idle & Processing Time Comparison SA Heuristic

### 5.5 Implementation Plan

This section will discuss an implementation plan for successfully implementing the designed solution model into Company X' daily operations. These implementation plan steps will elaborate on the steps required to use the scheduling method successfully in Company X' operations.

#### Optimise Computation Time

As discussed in Section 5.4, the computation time can be incremental when using the suggested input settings for the simulated annealing heuristic. To speed up this process, the scheduling model should be revised to remove unnecessary computations or database accesses, reducing the computational time required.

#### Modify Mendix Application

Currently, the Mendix application schedules the service orders every week but runs over the historical data and repeats the scheduling process for each week. To implement the Mendix application into operations, the model should be triggered once a week at the start of the week to develop the schedule.

#### Integrate Mendix Application

The Mendix application built for the scheduling method should be integrated with Company X' current ERP system. This should ensure that the incoming service orders are scheduled weekly, thus running a weekly scheduling instance. Furthermore, the modified Mendix application should be deployed and hosted by Company X ERP.

#### Improve Application

The scheduling application doesn't consider cybersecurity. Therefore, security measures should be taken to secure it. One example of these measures is requiring a login before accessing the application.

#### Mechanic Accessibility

Most importantly, mechanics should be able to access the service schedule and read all necessary information regarding a service order. To ensure proper use of the schedule, a user interface should be built to give the mechanics access to the required information and input comments regarding work performed on the service order. This opens up possibilities for further analysis of the service times.

## 5.6 Summary

This chapter has evaluated the solution model by first assessing the input settings used for the SA metaheuristic. Then, the input settings were used to determine the results of each scheduling methodology, which were evaluated. Lastly, an implementation plan is formulated to implement the scheduling model into daily operations. The research question can be answered:

*What recommendations and conclusions can be made after implementing the servicing methodology and schedule?*

#### Results from evaluation

This chapter evaluates both constructive heuristics and improvement heuristics to determine the best improvement heuristic that can be used to minimize the objective function and total makespan (in minutes). The results indicate that among constructive heuristics, the LPT heuristic was the best approach to scheduling all service orders for 2023, having 3.81% lower makespan than the current FCFS scheduling method. Afterwards, the improvement heuristics were evaluated and compared to the FCFS heuristic. This showed that the FCFS + SA heuristic has the best improvement over the FCFS heuristic of 8.38%.

Furthermore, all improvement heuristics were compared to their constructive counterparts, which showed that the FCFS and SPT heuristics performed better than the LPT heuristics, resulting in a higher makespan in some weeks. In conclusion, the FCFS + SA heuristic performed the best in terms of the completion time KPI.

Section 5.1.2 mentions that the idle time only considers days when at least one job is executed. These results are shown in Figures 22 and 27. However, days when no single job is performed are not considered. Figure 21 reveals that the scheduling model does not fill every day of the week. Only a few weeks reach over the 2000-minute mark for completion time, far below the workshop's workable minutes when five workable days are considered (2400 minutes), as discussed in Section 4.1.4. This means the scheduling model's input, the historical service order data 2023, does not reflect reality entirely correctly.

# 6 Conclusion & Discussion

This chapter will conclude the research conducted on the case of Company X. Section 6.1 will discuss the main findings and motivate the answer to the main research question from Section 1.4. Next, Section 6.2 will discuss recommendations made towards Company X. Section 6.3 will discuss possible further research topics for Company X. Then, Section 6.4 will reflect on the theoretical and practical contributions this research has brought. And lastly, Section 6.5 will discuss limitations and reflect on the scheduling model.

## 6.1 Conclusion

This section will discuss the main findings and motivate the answer to the main research question, which is stated as follows:

*What is the most effective approach to developing an improved service schedule method that Company X can use to improve the maintenance process and reduce service times?*

The answers to these main research questions can be found in the answers to each sub-research question included in each of the consequent sections. Section 2 performed a contextual analysis regarding the overall service process. All facets of the servicing process were analysed, from when a customer comes into contact with Company X to when they can use their equipment again. A BPMN was created to get a structured overview of how the service process was strictly operated, followed by an analysis of service times, which would be necessary for inputting the scheduling model. Also, the current scheduling method was analysed and written to be tested against the formulated scheduling model.

Next, Section 3 provided a theoretical framework to find suitable theories to find the best solution method for the case of Company X. This was done by performing a literature study to identify relevant theories and work towards identifying the scheduling problem best suited for the maintenance department. The theoretical framework also provided relevant approaches to perform a scheduling method; from the literature study, it was concluded that the maintenance department best suited an identical multiple-machine scheduling problem, and it could be best solved using a combination of heuristics, including the Simulated Annealing metaheuristic.

In Section 4, a solution method is formulated to be applied to the scheduling problem of Company X. Several assumptions had to be made for this scheduling model, and some constraints had to be considered. The objective function on which the improvement heuristic is focused is discussed. Furthermore, the requirement of the solution model is discussed before discussing the heuristics in more detail. The heuristics and scheduling model are discussed in detail to describe all aspects of the scheduling model. All important microflows that determine the service schedule are discussed. Lastly, the following section outlines how the KPIs are computed and used to evaluate the solution method.

Section 5 discusses the evaluation and implementation of the solution method. First, experiments were conducted to evaluate the best settings for the simulated annealing metaheuristic. This SA metaheuristic requires several input parameters for the improvement heuristic's cooling schedule and iteration length. These parameters were evaluated based on completion time and computation time. The completion time (or makespan) was used to assess which input settings performed best for the scheduling model, and the computation time was used to consider the computational efficiency of the scheduling model, as increasing computation time sometimes does not outweigh the benefits in makespan. The results from the iteration length experiment in Table 7 show that this occurs when an iteration length of thirty or fifty is used, and computation time increases drastically with no to little positive effect on the total makespan. An experiment was carried out for each of the four input parameters $T_{start}$, $T_{stop}$, $\alpha$ and $k$. Each experiment grabbed a range of possible values for the parameter and evaluated based on the two criteria. The experiments resulted in the following input parameters:

- Starting temperature $T_{start} = 500$
- Stopping temperature $T_{stop} = 0.01$
- Decreasing factor $\alpha = 0.97$
- Iteration length $k = 20$
- Stopping criteria $n = 250$

Next, constructive heuristics concerning the current scheduling method were evaluated. This showed that the SPT heuristic performed 1.14% worse on daily average idle time per mechanic and 2.72% worse on the total makespan compared to the current FCFS scheduling method. This evaluation showed that the LPT heuristic performed 10.47% better on the daily average idle time per mechanic and 1.20% better on the total makespan than the FCFS method. Lastly, the determined input settings were employed in the SA metaheuristic evaluation. The constructive heuristics were combined with the SA metaheuristic to determine the best-performing scheduling method. This showed that the FCFS + SA heuristic performed the best according to total makespan, having an 8.33% improvement over the FCFS scheduling method, followed by a 5.39% and 5.89% improvement of the SPT + SA and LPT + SA heuristic over the FCFS scheduling method, respectively. The SPT + SA and FCFS + SA heuristics showed similar results on idle time per mechanic, but the LPT + SA heuristic showed a 10.45% improvement over the FCFS scheduling method. Furthermore, the FCFS + SA, SPT + SA and LPT + SA heuristics showed a 7.22%, 5.39% and 5.89% improvement on total makespan over the LPT heuristics, respectively. While the LPT + SA heuristic gave a similar idle time to the LPT heuristic, the FCFS + SA and SPT + SA heuristic showed a 12.09% and 13.22% increase in idle time over the LPT heuristic, respectively.

Overall, the FCFS + SA heuristic performed the best in determining a scheduling method, resulting in the lowest total makespan. However, it should be noted that all heuristics showed that several days remained empty after scheduling all jobs based on historical data from the ERP. This could be due to incorrect regular service job times or incorrect service times registration. A single service order's service time includes two aspects: regular service jobs and 'extra' service time, as discussed in Section 2.3.3. This extra service time is considered when the maintenance doesn't fit in one of the service jobs from Table 2. The regular service job times are taken from interviewing all involved employees from the servicing process. However, errors may be made in these times, which could cause emptiness in the scheduling model. This emptiness could also be caused by possible incorrect registration of service hours, which will be discussed in Section 6.5.

The LPT + SA heuristic performed best regarding the daily average idle time per mechanic, directly related to the better-performing LPT heuristic compared to other constructive heuristics. However, it should be noted that the average daily idle time per mechanic doesn't consider days where no jobs are scheduled, which means that the idle time is even higher. This is reflected in the reasonably high idle time ratios, which also don't consider the days when no jobs were scheduled, meaning that the actual idle time ratio is even higher. Possible causes of this will be discussed in Section 6.5, and recommendations will be made in Section 6.2.

Considering all sections, the main research question is answered by combining the current scheduling method with the Simulated Annealing metaheuristic to develop an improved service schedule, which can be used to improve the maintenance process and reduce overall service time. However, other heuristic combinations shouldn't be left out, as these could be effective when the emptiness of the scheduling model is reduced.

### 6.2 Recommendations

From the conducted research, several recommendations can be drawn to advise Company X.

First, the maintenance department should start implementing a more dedicated scheduling policy. All service orders are processed first-in-first-out, but no policy is in place to manage the schedule. Therefore, it is recommended that the mechanics who service the equipment start using a policy that is also complied with.

Another recommendation can be drawn from the need to implement such a scheduling policy. A scheduling application should be made so all stakeholders (mechanics, customer service, management, and possibly customers) can see and edit the service schedule and service orders involved. This application should integrate the simulated annealing scheduling method to schedule the service orders appropriately. The application should also have different permissions for different stakeholders. For example, a mechanic should only be able to see the schedule and make annotations but shouldn't be able to change the schedule as a whole.

As discussed in Section 6.1, the service schedule had a lot of emptiness. Therefore, it is recommended that we investigate what is causing this deviation from reality. Investigation should be started into how the hours are registered on the invoices, and changes should be made accordingly; if a scheduling application is implemented, these hours can be registered automatically by having a mechanic register when he starts and finishes a service order.

### 6.3 Further Research

This research has developed a Mendix scheduling model that determines the best schedule for the maintenance department. However, the solution model is not optimal, and further research is possible.

The Mendix application itself can be optimised to calculate the best schedule. Section 5 discusses the computationally extensive Mendix application; there is still ground to be gained on the computational efficiency of the scheduling model.

Furthermore, research can be conducted into service times for each categorised service item. Each service item was depicted in a category with associated service times. However, these service times are merely the average of a category's service time. Further investigation into these service times could be fruitful as this could be incorporated into the scheduling model, anticipating the time needed for a service order. This research could benefit the maintenance department's overall service level and enhance the maintenance process.

Seasonality in the demand for specific service categories can be researched as well. As described in Section 2, several recurring jobs must be performed for Company X clientele. Further research on the demand for these orders could have cost benefits and optimise resource allocation.

Currently, the days when service is performed outside the workshop are not taken into account while scheduling (please refer to Section 2.6 for more details). The scheduling model can be improved by conducting further research in this category. The scheduling model should consider the planning and overall process involved in performing on-site maintenance. To make the scheduling model more realistic, research should be conducted to consider breaks and other time-consuming activities, which are not currently considered in the scheduling model.

### 6.4 Contributions

This section discusses both the theoretical and practical contributions of this research.

### 6.4.1 Theoretical Contribution

The theoretical contribution of this research is on the topic of job scheduling problems. Traditionally, most cases use scheduling difficulties, and heuristics are based in sectors like production systems and logistics. This research used these heuristic approaches in a different sector, namely the service sector. This research shows that the heuristic approaches are multi-interpretable and can be used in various situations and sectors.

### 6.4.2 Practical Contribution

This research has practically contributed to the company it is conducted at, Company X. The practical contributions lie in Section 4, where a solution model is designed and formulated; the solution model is developed in Mendix, an application Company X is already using today. Therefore, the integrability of the solution model is excellent, and following the implementation plan in Section 5, the solution model can be implemented into a scheduling application for Company X.

## 6.5 Limitations

This section will cover the limitations of the developed scheduling method and any further discussion or limitations regarding this research. These should be considered when implementing the technique into daily operations.

Due to the wide variety of service items processed by the maintenance department, no statistical distributions could be analysed. The service orders were categorised, which weren't extensive enough to depict any statistical distributions for the categorised service times. Next, the processing times for the standard service jobs were analysed by interviewing the workshop mechanics. These processing times, therefore, cannot be depicted as exact. The average for these categories was used as processing times, which leads to a certain degree of uncertainties in the scheduling model. These uncertainties, therefore, limit the reflection of reality and do not represent real life well enough.

As discussed in Section 5.6, the scheduling model didn't fill the schedule well enough, leading to empty days, where no jobs were performed, and high idle times for the days that were filled. This was caused by a situation where, in most weeks, only two to three days could be filled with scheduled jobs. Several factors can cause this. Section 2.3.3 describes that the processing times are analysed in mechanic interviews and data analysis. The first analysis to determine the standard service jobs was conducted by interviewing the mechanics. These interviews were held to understand better the overall servicing process and the times taken for these standard service jobs. It is possible that the mechanics misestimated the service times, leading to lower service times than the real-life situation. Furthermore, the categorised service times were analysed through data analysis. Several categories were formulated to understand the service times better. Each category was analysed based on averages, maxima, minima, and service occurrences. The averages of these categorised service times were also used as input for the scheduling model, which can also have caused the emptiness in the scheduling model. The times used for the data analysis come directly from the ERP system; these times are stated on the invoice sent to the customer as hours worked on a particular service order. It is possible that these times are systematically incorrectly determined and put on the invoices, leading to the emptiness in the scheduling model. This could also have significant consequences for the overall revenue of Company X, as this would mean that the customers aren't charged enough hours for the work performed by the mechanics.

Lastly, the steps and processes taken in this bachelor thesis are practically relevant to the case of Company X as input is directly taken from their operations, and results are based on these operations. Therefore, the process formulated in the preliminary chapters cannot be seen as a framework for implementing such a schedule improvement model. However, as discussed in Section 6.4, the use of heuristics to solve scheduling problems is mainly applied in the production and logistics sectors. This research can be seen as an exploration of applying job scheduling to new industries that traditionally haven't been considered. Therefore, the process can be followed, and a similar path can be used in different situations.

As discussed in Section 3, the job allocation problem is viewed as a job scheduling problem. However, the job allocation problem can also be viewed as a personnel scheduling problem. In this problem, employees are assigned to shifts or tasks while considering specific objectives and constraints, such as minimizing labor costs or workforce utilization while adhering to labor regulations and shift preferences. [29] Although this type of resource allocation problem may seem useful for the workshop schedule at Company X, it is not a viable option. The availability of mechanics varies weekly, but usually, all three mechanics are present at the workshop during working days. Since there are only three employees in the workshop, it is not practical to use personnel scheduling to manage the workshop schedule, as there the possible gain by using a personnel scheduling problem is limited.

# References


[1] Xiong, H., Shi, S., Ren, D. & Hu, J. A survey of job shop scheduling problem: The types and models. *Computers & Operations Research* **142**, 105731 (2022).

[2] Fakhravar, H. Combining heuristics and exact algorithms: A review. *arXiv preprint arXiv:2202.02799* (2022).

[3] Heerkens, H. & van Winden, A. *Solving Managerial Problems Systematically* (Noordhoff Uitgevers, 2017). Translated into English by Jan-Willem Tjooitink.

[4] Saunders, M. N., Lewis, P. & Thornhill, A. *Research methods for business students* (Pearson, 2019), eigth edn.

[5] Marques, P. A. & Requeijo, J. G. Sipoc: A six sigma tool helping on iso 9000 quality management systems. In *XIII Congreso de Ingeniería de Organización*, 1229–1238 (2009).

[6] Kocbek, M., Jošt, G., Heričko, M. & Polančič, G. Business process model and notation: The current state of affairs. *Computer Science and Information Systems* **12**, 509–539 (2015).

[7] Baker, K. R. Introduction to sequencing and scheduling. *(No Title)* (1974).

[8] Baker, K. R. & Trietsch, D. *Principles of sequencing and scheduling* (Wiley-Blackwell, Hoboken, NJ, 2009).

[9] Graham, R., Lawler, E., Lenstra, J. & Kan, A. Optimization and approximation in deterministic sequencing and scheduling: a survey. In Hammer, P., Johnson, E. & Korte, B. (eds.) *Discrete Optimization II*, vol. 5 of *Annals of Discrete Mathematics*, 287–326 (Elsevier, 1979). URL https://www.sciencedirect.com/science/article/pii/S016750600870356X.

[10] Chen, B., Potts, C. N. & Woeginger, G. J. A review of machine scheduling: Complexity, algorithms and approximability. *Handbook of Combinatorial Optimization: Volume1–3* 1493–1641 (1998).

[11] Martello, S. & Toth, P. Algorithms for knapsack problems. *North-Holland Mathematics Studies* **132**, 213–257 (1987).

[12] Cacchiani, V., Iori, M., Locatelli, A. & Martello, S. Knapsack problems — an overview of recent advances. part i: Single knapsack problems. *Computers Operations Research* **143**, 105692 (2022). URL https://www.sciencedirect.com/science/article/pii/S0305054821003877.

[13] Nicholson, T. & Pullen, R. A permutation procedure for job-shop scheduling. *The Computer Journal* **11**, 48–56 (1968).

[14] Laporte, G. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* **59**, 345–358 (1992). URL https://www.sciencedirect.com/science/article/pii/037722179290192C.

[15] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. & Shmoys, D. B. Chapter 9 sequencing and scheduling: Algorithms and complexity. In *Logistics of Production and Inventory*, vol. 4 of *Handbooks in Operations Research and Management Science*, 445–522 (Elsevier, 1993). URL https://www.sciencedirect.com/science/article/pii/S0927050705801896.

[16] Pinedo, M. L. *Planning and scheduling in manufacturing and services* (Springer, 2009).

[17] Elibol, A. *et al.* A single machine job scheduling problem with sequence dependent setup times. In Durakbasa, N. M. & Gençyılmaz, M. G. (eds.) *Digital Conversion on the Way to Industry 4.0*, 804–816 (Springer International Publishing, Cham, 2021).

[18] Rutenbar, R. Simulated annealing algorithms: an overview. *IEEE Circuits and Devices Magazine* **5**, 19–26 (1989).

[19] Hancerliogullari, G., Koksalmis, E. & Hancerliogullari, K. O. *Heuristics and Simulated Annealing Algorithm for the Surgical Scheduling Problem*, 225–241 (Springer International Publishing, Cham, 2016). URL https://doi.org/10.1007/978-3-319-26024-2_12.

[20] Banerjee, A., Singh, D., Sahana, S. & Nath, I. Chapter 3 - impacts of metaheuristic and swarm intelligence approach in optimization. In Mishra, S., Tripathy, H. K., Mallick, P. K., Sangaiah, A. K. & Chae, G.-S. (eds.) *Cognitive Big Data Intelligence with a Metaheuristic Approach*, Cognitive Data Science in Sustainable Computing, 71–99 (Academic Press, 2022). URL https://www.sciencedirect.com/science/article/pii/B978032385117600008X.

[21] Brandimarte, P. Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research* **41**, 157–183 (1993). URL http://dx.doi.org/10.1007/BF02023073.

[22] Kirkpatrick, S., Gelatt Jr, C. D. & Vecchi, M. P. Optimization by simulated annealing. *science* **220**, 671–680 (1983).

[23] Lawler, E. *The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Wiley-Interscience series in discrete mathematics and optimization (John Wiley & Sons, 1985). URL https://books.google.nl/books?id=qbFlMwEACAAJ.

[24] Talbi, E.-G. *Metaheuristics: from design to implementation* (John Wiley & Sons, 2009).

[25] Koulamas, C. Decomposition and hybrid simulated annealing heuristics for the parallel-machine total tardiness problem. *Naval Research Logistics (NRL)* **44**, 109–125 (1997).

[26] Vallada, E., Ruiz, R. & Minella, G. Minimising total tardiness in the m-machine flowshop problem: A review and evaluation of heuristics and metaheuristics. *Computers Operations Research* **35**, 1350–1373 (2008). URL https://www.sciencedirect.com/science/article/pii/S0305054806002048.

[27] Holthaus, O. & Rajendran, C. Efficient dispatching rules for scheduling in a job shop. *International Journal of Production Economics* **48**, 87–105 (1997). URL http://dx.doi.org/10.1016/S0925-5273(96)00068-0.

[28] Busetti, F. Simulated annealing overview. *World Wide Web URL www. geocities. com/francorbusetti/saweb. pdf* **4** (2003).

[29] Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E. & De Boeck, L. Personnel scheduling: A literature review. *European Journal of Operational Research* **226**, 367–385 (2013). URL https://www.sciencedirect.com/science/article/pii/S0377221712008776.

# Appendices

## A Completion Time Comparison

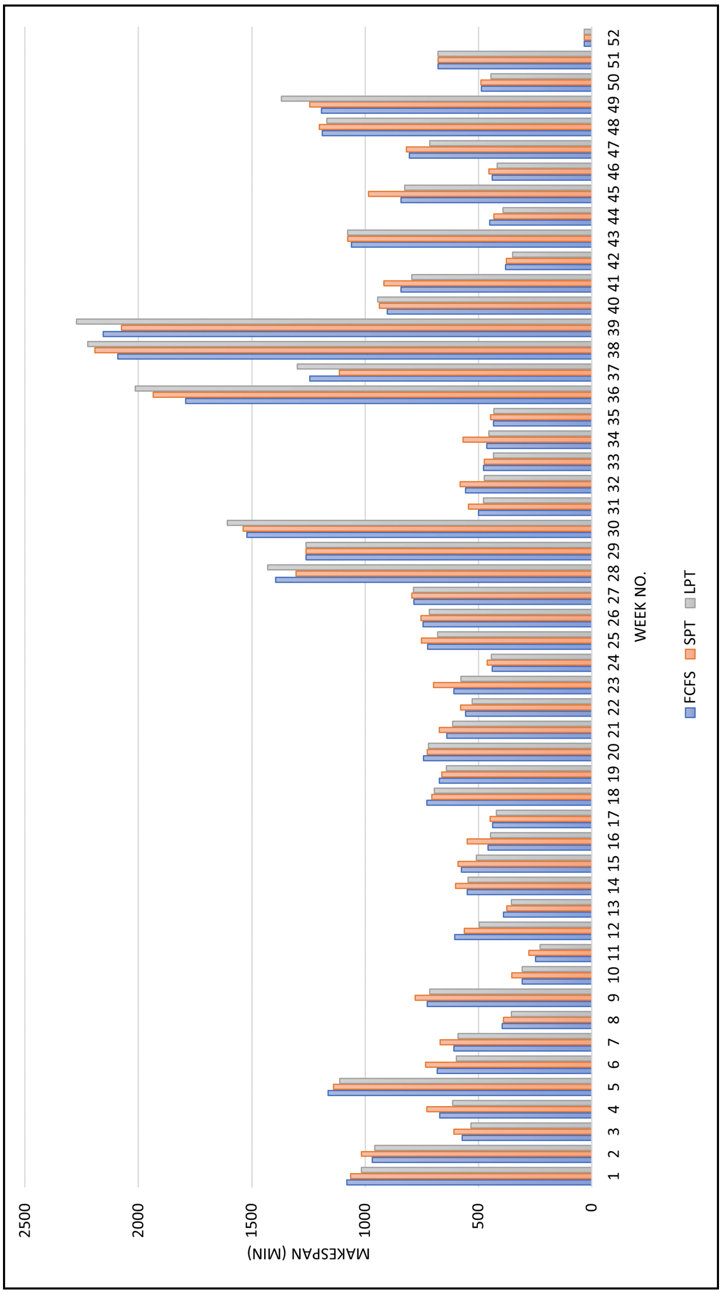


Figure 21: Comparison Completion Time

# B Idle Time Comparison

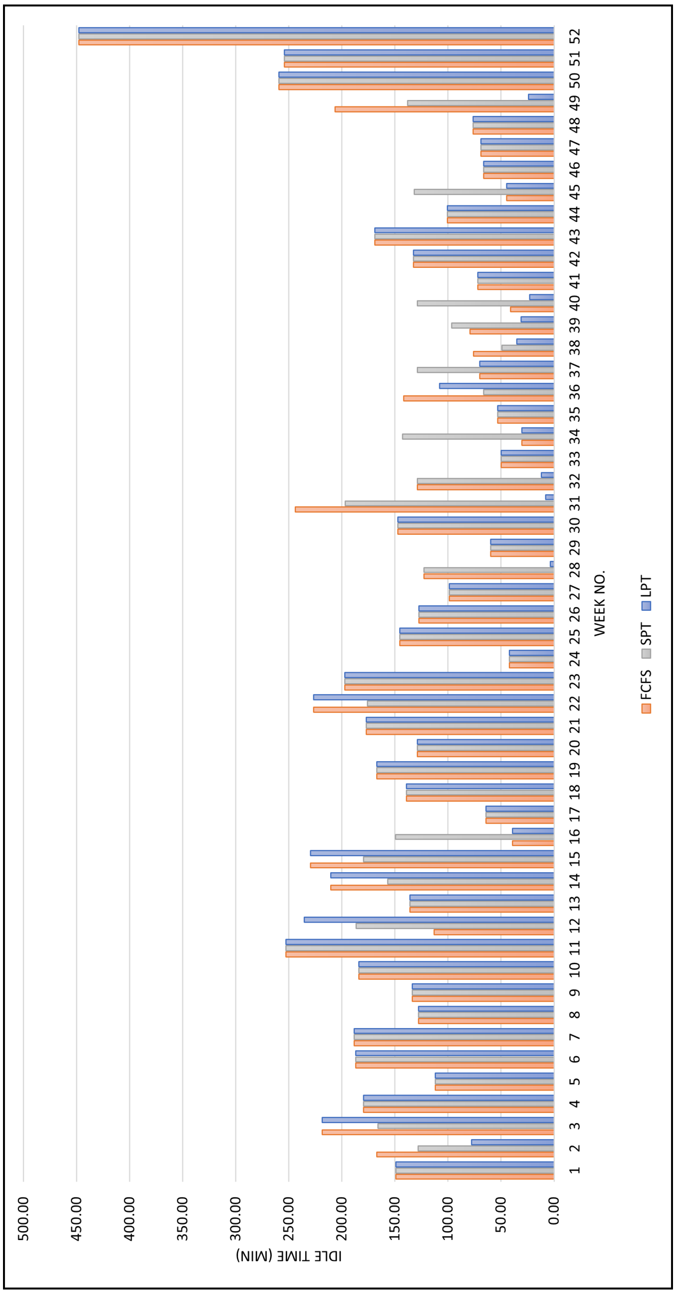


Figure 22: Comparison Average Idle Time

# C Completion Time Comparison SA

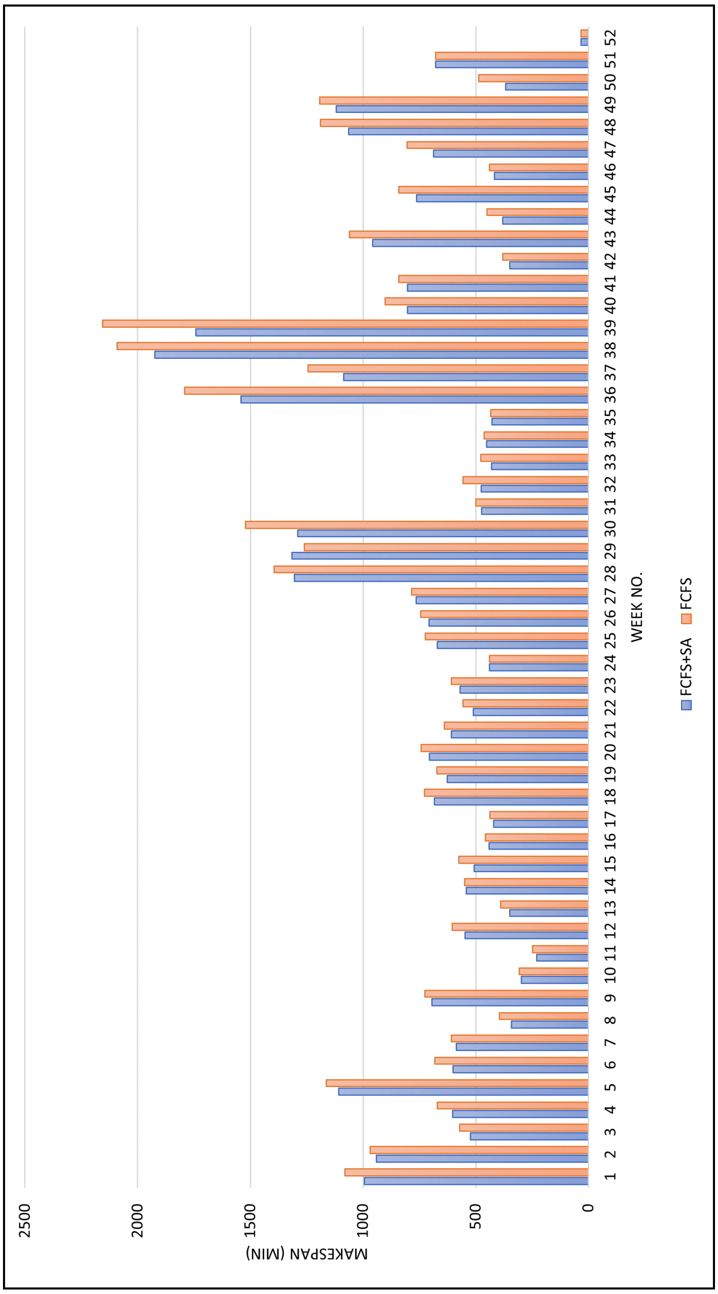


Figure 23: Comparison Completion Time FCFS

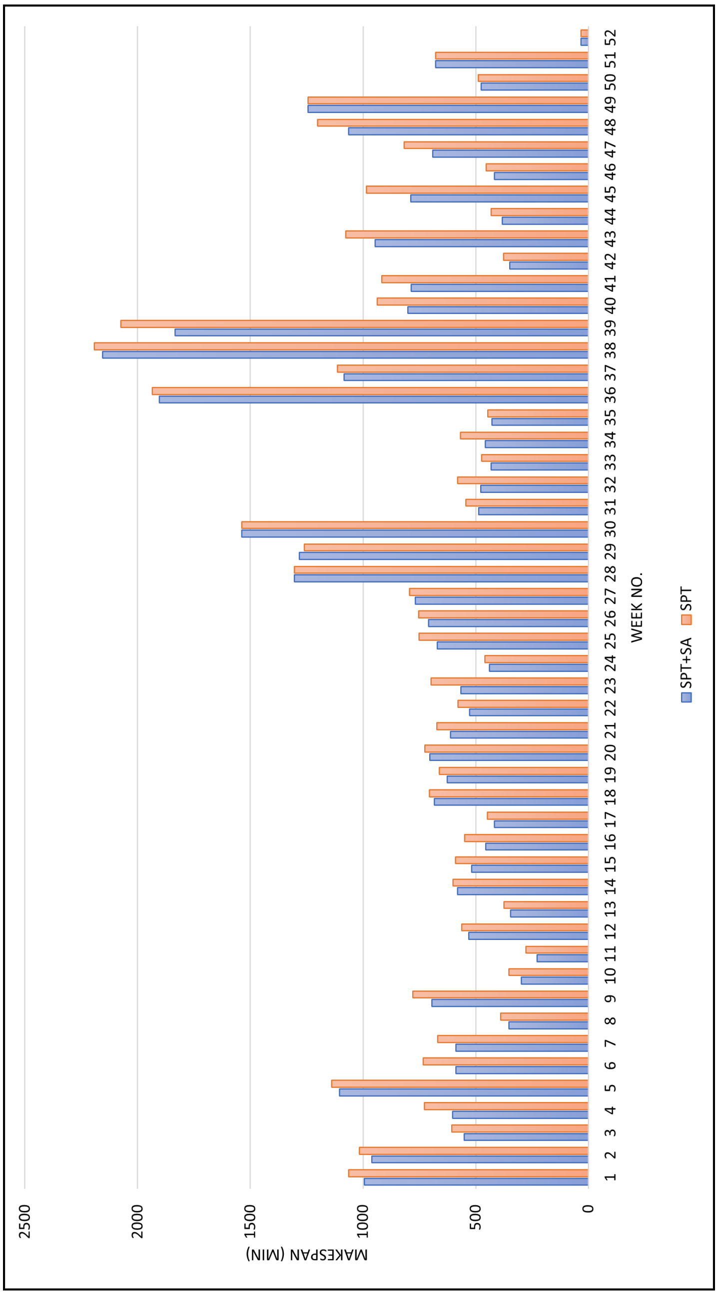


Figure 24: Comparison Completion Time SPT

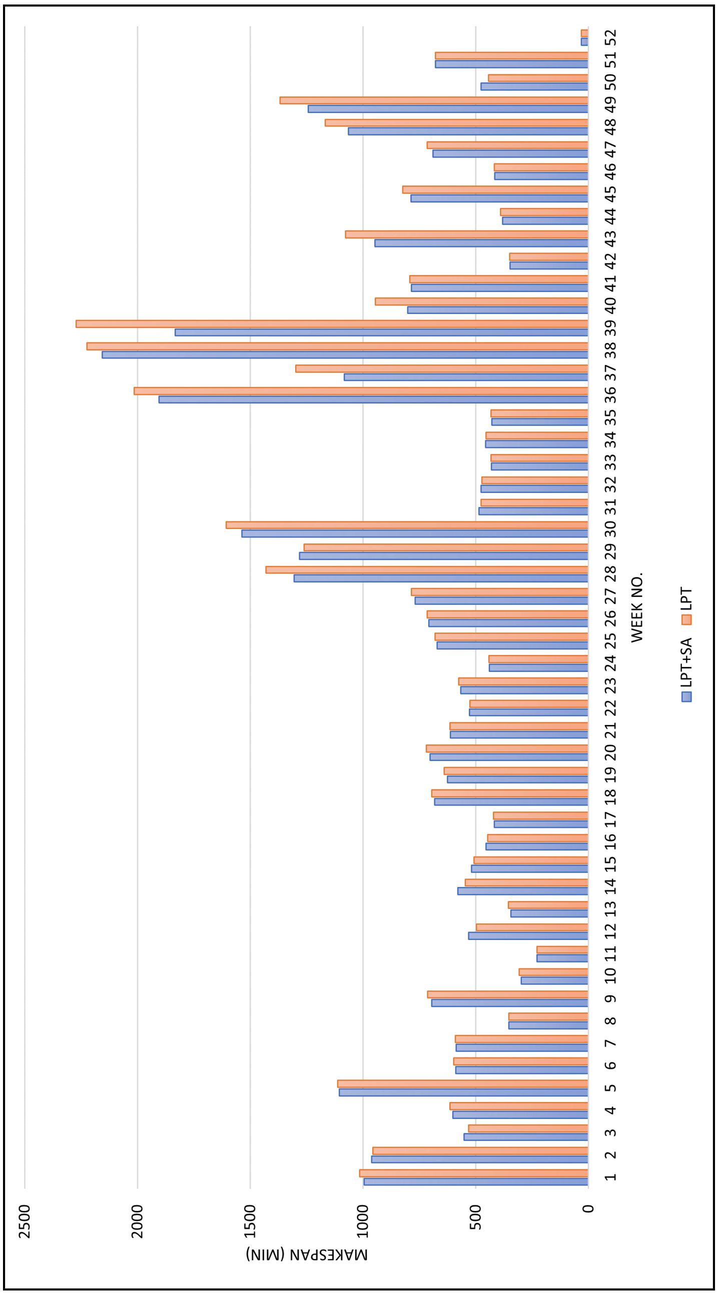


Figure 25: Comparison Completion Time LPT

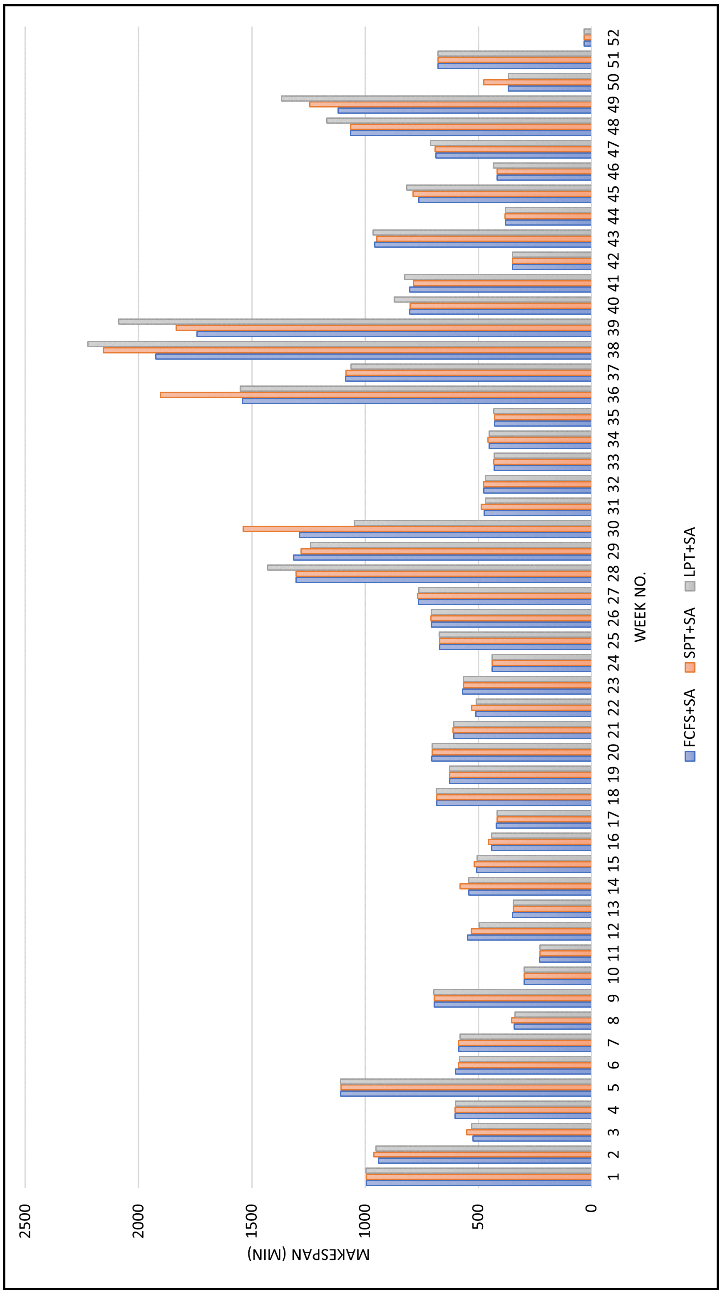


Figure 26: Comparison Completion Time SA

# D Idle Time Comparison SA

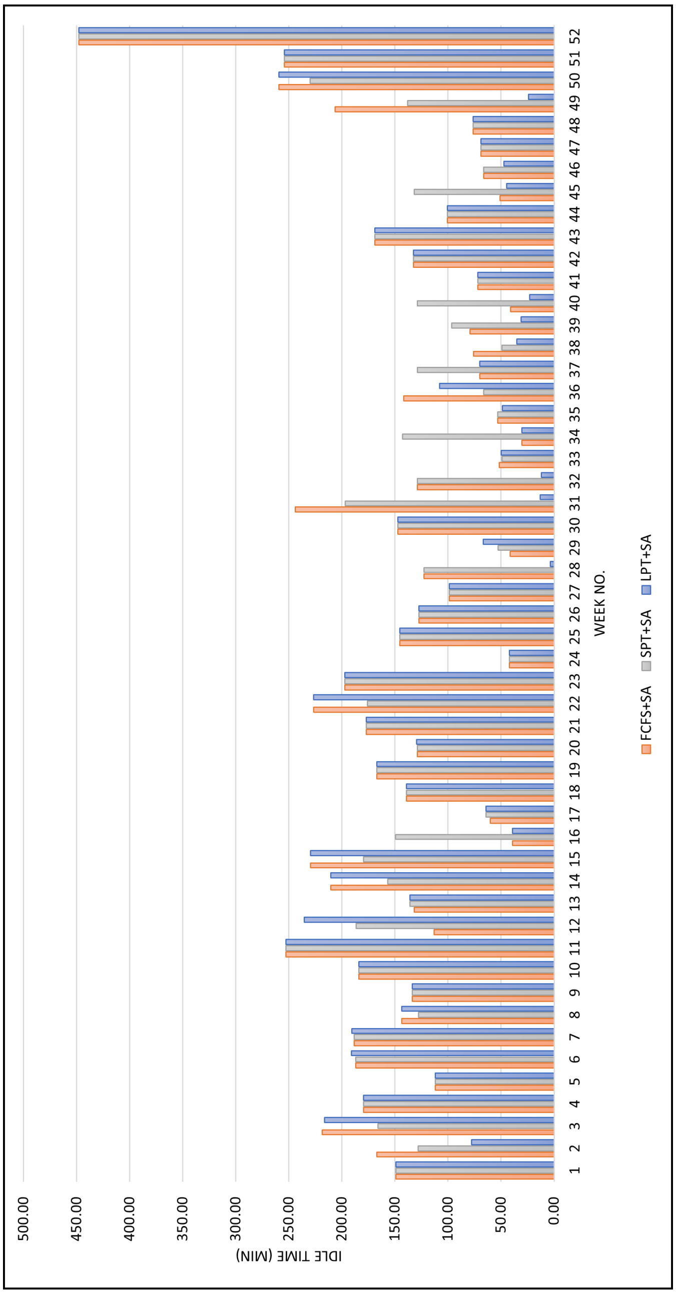


Figure 27: Comparison Average Idle Time SA