UNIVERSITY OF TWENTE.

UNIVERSITY OF TWENTE.

FACULTY OF BEHAVIOURAL, MANAGEMENT AND SOCIAL SCIENCES (BMS)
INDUSTRIAL ENGINEERING AND MANAGEMENT

Improving the production schedule and scheduling process of ABI

Mark Huizinga

First supervisor: Marco Schutten
Second supervisor: Eduardo Lalla
Company supervisor: Henk Slettenhaar(ABI)

ABI MAT BRUSH since 1910

Preface

Dear reader,

You are about to read my thesis on "Improving the production schedule and scheduling process of ABI". As the title suggests, the research was performed at the Dutch company ABI. In the past time, I have gained tremendous insight into the operations, experience, and atmosphere at ABI. This experience also allowed me to get more insight into subjects discussed broadly in the university of Twente study program of Industrial Engineering and Management.

I would like to thank everyone from ABI, for always being available for questions, feedback, and the nice work atmosphere they have at the office. Especially, I would like to thank Henk for being patient and allowing me to fall and, help me get up again after we hit a dead end in the research. Secondly, I want to thank my supervisor Marco Schutten for always providing me with helpful feedback, whilst letting me choose my path in the research. Eduardo Lalla I would like to thank for providing me with feedback that helped create a more coherent story in my research. At Last, I would like to thank all of my friends and family that supported me during the time I worked on my thesis.

I hope you enjoy my thesis and gain some knowledge on the topics we discussed.

Best wishes,

Mark Huizinga

Management summary

Introduction

ABI is a production company that produces entrance mats and technical brushes. This research only regards the technical brushes part of the company. ABI has difficulties communicating accurate delivery times to their customer. The main cause of this problem is that ABI is unable to make a production schedule. This has a few causes, the production schedule is made manually, the sequence of the remaining orders is very flexible, and there is no information about the remaining time orders need to be produced. Also, due to an increase in incoming orders, the standard delivery time of 3 weeks cannot be met, leading to a lot of backorders. In this thesis, we research the production schedule and scheduling process of ABI to improve their production schedule. After this research ABI has method to create a production schedule with more information available about the sequence and production times of the orders. ABI can use the production schedule to communicate more accurate delivery times to their customers.

Approach and Results

To improve the production schedule of ABI some steps need to be taken. First, we map out the production process to investigate which stage is the most important. Secondly, we look into the current way ABI schedules the orders, to find out how that process can be improved. Thirdly, we choose a scheduling method that is suitable for ABI and test this scheduling method. We test the scheduling methods by performing experiments on data acquired from ABI.

We investigate multiple scheduling problems with their different characteristics to discover how the scheduling situation of ABI can be classified. Based on our findings this scheduling problem can be classified as a single-machine scheduling problem. In this situation, there are a number of orders that need to be scheduled on 1 machine. To solve this scheduling problem different methods can be used, we use priority rules. Priority rule-based scheduling consists of 2 parts: *the generation scheme* and the *priority rule*. The generation scheme can be distinguished as serial or parallel. For ABI the parallel generation scheme is better because there is more idle time with the serial generation scheme. We choose the priority rules Earlies Due Date(EDD) and Minimum Slack(MinSlack). Both of these priority rules use the due date to determine the *priority value* of an order. For ABI the setup time is also an important factor, so we included the setup time in the priority rules.

However, the schedule resulting from a priority rule is not the optimal solution. To improve the schedule made with the priority rules we use the Steepest Descent method. With this method, all of the *neighbor solutions* are created and the best solution is chosen as the new solution. All of the *neighbor solutions* for the new solution are considered, again the best solution is taken as the new solution. This process repeats itself until there is no better solution. A neighbor solution is a solution that is created by making small changes to the initial solution, in our case by swapping 2 orders with each other. In our research we integrate the Key Performance Indicators(KPIs) average completion time with the maximum lateness into 1 KPI to determine what neighbor solution is better. This KPI is calculated in the following way:

$$KPI = Average\ Completion\ Time + Maximum\ Lateness$$

We model the scheduling situation to implement and test the priority rules and the improvement method. The most important stage in the production process is the drilling and punching stage, this is the reason we model this stage. The drilling and punching stage is the most important stage in the production process because: it is the bottleneck of the production process, it is the only stage with setup times, all of the products need to go through this stage, and lastly, the processing times can be estimated in this stage, which is important to know in scheduling the orders. To compare and measure the quality of the schedules made with the 2 priority rules with each other and the effect of

the Steepest Descent method we use 3 KPIs. These KPIs are the average completion time, maximum lateness, and average lateness.

We test the influence of the setup time within the priority rules. We do this by multiplying the setup time by a weight when calculating the priority values of the orders and looking at the change in the KPIs. Based on the experiment we did not find a value for the weight that resulted in the best schedule for all of the data sets we used. Also, we cannot determine which priority rule is better, it depends on the data set which priority rule performs better. However, by increasing the value for the weight of the setup time, based on the 3 KPIs the quality of the schedule can increase.

We test the improvement method Steepest Descent by comparing the values of the KPIs after using the priority rules with the values after using the Steepest Descent method. The average relative change for implementing the Steepest Descent method on the MinSlack rule is for the KPIs average completion time -3,97%, maximum lateness -34,41%, and, average lateness -65,34%. For the EDD rule this is -2,71%, -28,55%, and, -28,31% respectively. Based on the values for the KPIs the EDD rule performs better with the KPIs average completion time and average lateness, while the MinSlack rule the value for the maximum lateness is better.

Conclusion and Recommendation

The conclusion from this research is that ABI can improve its production schedule and a scheduling method to provide their customer with more accurate information about their order delivery time. We recommend using the priority rules EDD and MinSlack with different weights for the setup time to find the best schedule based on the values of the KPIs average completion time, maximum lateness, and average lateness. We recommend using the Steepest Descent method to optimize the schedule, with regard to the average completion time and maximum lateness.

Contents

1. Introduction

Section 1.1, provides background information about the company ABI that is subject in this research. *Section 1.2,* describes the management problem and in *Section 1.3* the problem cluster reveals the core problem. *Section 1.4* states the research question, together with the sub-questions, and how they are answered. In *Section 1.5,* we discuss the scope of this research.

1.1 The Company

ABI is a company located in Almelo and founded in 1910. ABI is a medium-sized business and is a production company. Their products are split up between entrance mats and technical brushes. In this research, the part of the company regarding the technical brushes is investigated. *Figure 1-1* depicts different types of technical brushes. The difference between normal brushes and technical brushes is that technical brushes are used in an industrial environment and are used for certain production processes. The technical brushes are specifically made with certain dimension tolerances, for example, minimal traction or to withstand static charge. Technical brushes can be used for a variety of things, including cleaning, transportation of goods, or positioning of (fragile) materials. ABI has two main departments, the office, and the production hall. The office is divided into sales for the entrance mats, sales for the technical brushes, and management. The production hall is divided into the production of the entrance mats, production of the technical brushes, packaging, and shipping of the products.

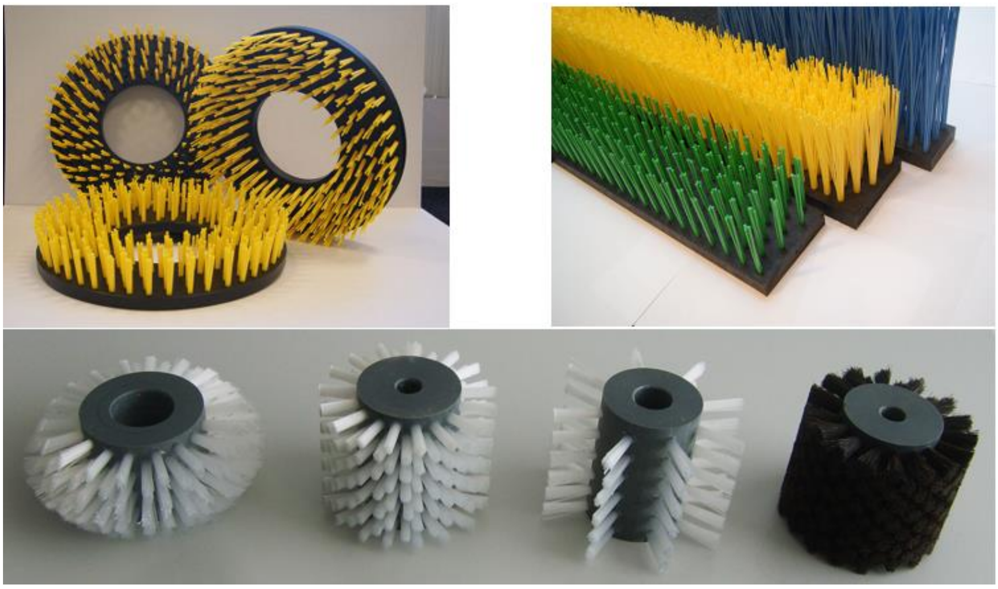

Figure 1-1: Discs(upper left), lats(upper right) and rollers(bottom)

1.2 Management Problem

ABI is a medium-sized company and the management problem is that there is no readily available information about the delivery time to communicate this with their customer. The main difficulty for ABI is to create a schedule that displays when an order is being processed and when it is expected to be delivered. The reason it is difficult is because there is no system in place to schedule the orders and that creates a good overview of the sequence of the orders that need to be scheduled. Variables like machine availability and delivery times of raw material are not considered. For instance, an order consists of a number of the same products, ranging from a couple to thousands of products. Different orders can differ a lot from each other with respect to product characteristics. For example, the length of the brush, the material of the brush, the width of the hole where the brush is put in, and the type of material and product in which the brush is put, which can be a lat, a disc, or, a roller. The discs and the lats are not in stock so they need to be ordered from a vendor, meaning the order cannot be produced right away. The order comes in, and based on the product's characteristics it is allocated to a machine through the employees' experience. When the order is allocated to a machine it is decided what steps need to be taken for these products to complete the order. These orders are put on a list and this list is sent to the production hall to be manufactured. The sequence in which these orders are produced is partly based on the date the order came in, so first-in-first out, and partly on the decision of the head of the production hall. This sequence is very dynamic, it is not certain what order will be scheduled to be produced in a couple of days. Another influence on the production schedule is that customers can call and ask whether their order can be rushed. The effect of this decision-making process regarding the production schedule is that the remaining orders are delayed. All these variables regarding the production schedule need to be considered. However, for ABI this is too much to keep track of, so, there is no clear sequence of the remaining orders. The problem that arises is that ABI cannot give information to the customer about when their order will be ready.

1.3 Problem Definition

In the starting situation, ABI does not have a production schedule, only an overview of the orders that still need to be produced from the previous weeks and the coming week. The process of how the remaining orders are scheduled is done manually. With the manual process, the standard delivery time of 3 weeks could be met; however, the number of orders increased, causing the number of backorders to increase. The overview list is reviewed by the head of production of the technical brushes with 35 years of experience and the office team. Together they decide the sequence of the orders in the coming week that need to be produced, keeping in mind the due dates of the orders and other variables that influence the efficiency of production, like when an order can be produced. Also, a customer can call and tell they need their order as soon as possible. Then, the sequence can be interrupted and the fast-tracked order is produced sooner than was planned initially. Fast-tracking an order cannot be done that easily, but it is possible against compensation or if the request is made by a regular customer. These factors lead to uncertainty with regard to the delivery times of the remaining orders and lead to delays. Therefore, ABI cannot provide their customer with the correct information about the delivery time of their order.

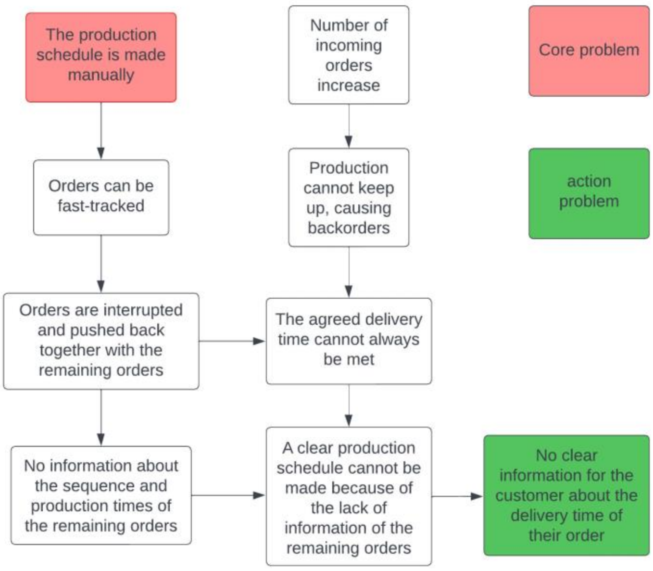

Figure 1-2: problem cluster of ABI

In *Figure 1*-2, the problem cluster of ABI is made with a clear overview of different problems and causes, that lead to other problems. The cause that ABI gets more orders is not a problem, but it causes problems. Not all incoming orders can be produced before the agreed deadline, leading to backorders. This means that the agreed delivery time cannot always be met. To solve this, more machines could be acquired and more people can be hired. However, this is quite cost-intensive and does not solve another problem that occurs, to make a clear production schedule. The production schedule is made manually and the people working on the orders have an influence on the sequence. Orders can be fast-tracked, this causes the remaining orders to be pushed back and produced at a later time. This shift in the sequence in the production process leads to uncertainty for the remaining orders that need to be produced. Because of these uncertainties and lack of information, ABI is unable to make a clear production schedule, causing there to be no accurate information about the delivery times of the remaining orders. Therefore, the action problem is that there is no clear information for the customer about the delivery time of their order. The core problem is that the production schedule is made manually.

1.4 Research Question and Research Design

To solve the selected core problem we formulate and answer a research question. To make the research question tangible and manageable, it is split up into sub-questions. When answering the sub-questions, knowledge is obtained to answer the research question.
The research question:
How can ABI get more insight into the production scheduling process to improve its production schedule?

Sub-questions 1:

- What is the current situation of ABI, regarding the production process and the production schedule?

It is important to know the company's current situation since it will reveal how ABI operates and give insight into variables that need to be taken into account when determining the best method to make the production schedule. The data-gathering method is descriptive as well as quantitative. Data needs to be gathered to understand the way ABI functions and what the characteristics are regarding the production and planning process. Also, data about the number of orders and backorders needs to be gathered to get a complete picture of what and how much is scheduled and produced each week. This research question is answered in Chapter 2.

Sub-question 2:

- What kind of machine scheduling problems situations are discussed in the literature, and how is a schedule made in those situations?

In Chapter 3 this question is answered by consulting the literature about scheduling. After the current situation is researched, information needs to be gathered on different scheduling situations and scheduling methods to solve scheduling problems, to implement the right scheduling method for ABI.

Sub-question 3:

- How can the scheduling situation and problem of ABI be classified, and how can this problem be solved?

In Chapter 4, the kind of machine scheduling situation and problem of ABI are classified, and a scheduling method is chosen to solve this problem. This is done by using the information from the current situation in Chapter 2, and the information gathered from the literature research in Chapter 3. The scheduling situation is described in a model, this model is used to implement different ways of scheduling and a method to improve an initial schedule.

Sub-question 4:

- What way of scheduling works the best for ABI, and how can the schedule be further improved?

In Chapter 5, this research question is answered by doing experiments to test the effectiveness of the different scheduling methods and test the effect of the improvement method. Based on these findings ABI can choose the scheduling method that works best for them.

1.5 Scope

The scope of this research is regarding the machines of ABI. Information needs to be gathered on the specifications like the processing times and what kind of products are produced on a particular machine. The operation of ABI has 2 kinds of operations, as mentioned in *Section 1.1*, this research is performed on the production of the technical brushes, since this is where the problems lay with respect to the production schedule. Regarding the technical brushes, we look into the automated machines, not products that are made with manually operated machines. The reason for only using the automated machines is because the manual machines' processing speed is too variable. This means only the scheduling of orders that contain the rollers and lats are investigated, since the discs are only made on the manual machines.

2. Current situation

In this chapter, the current situation of ABI is described. In *Section 2.1*, we discuss the company structure. *Section 2.2,* provides information about the machines that are used in the production process and activities that happen at those machines. *Section 2.3* gives information about the products that ABI produces. Then, we map the production process of ABI in *Section 2.4*. *Section 2.5* describes how orders are scheduled to be produced. In *Section 2.6*, we discuss the data that is available and how we use the data. *Section 2.7* gives a conclusion of the chapter and an answer to the research question related to this chapter.

2.1 Company Structure

In this research, only the departments related to technical brushes are discussed. ABI has essentially two big departments which are divided into smaller departments. One of the departments is the production hall, which takes care of the production of the technical brushes. Which activities and roles consist in the production hall are further explained in *Section 2.4.* The second is the office department, this department consists of sales, management, human resources, and accounting. Sales register the orders of the customers and make sure the employees working in production have all the information they need to make the product.

2.2 Machines

ABI uses different machines to produce its products: *the lathe, the drilling, and punching machines*, and *the cutting machines*. At the lathe, the raw material of the rollers is prepared for the drilling and punching machine. Here, the material is sawed to size, and if needed specific holes are drilled. There are 5 different drilling and punching machines used in the production. The pace of the drilling and punching machine determines the production speed. The machines work as follows. Before any material is put in, the machine needs to be set up. There is a general setup time for each order that needs to go on a machine. Different dimensions of the products need to be set up in the machine. There are different speeds concerning specific patterns of the holes in the raw material, space in between holes, and what kind of brush material is used. Another setup time occurs when a different *head* is needed on the machines to drill a different size hole in the material and to fill that hole with the brush material. The head size can be the same between different orders, so this setup time is not always needed. Both setup times take 30 minutes. In every drilling and punching machine, there are 2 drills and 1 punching arm. The raw material is put in the machine to be drilled, and when it is done it is moved up or down to the punching arm. In *Figure 2-1,* it is depicted in the top part that the holes are drilled into the raw material, and in the bottom part of the figure, the brushes are punched into the material. After the brush material is punched into the product it is manually removed from the machine, new material is manually put in to be drilled. The 2 drills take turns, but the punching arm is always active. In *Figure 2-2*, the punching of the brush material into the product is depicted more clearly. There are also *manual punching machines* for each type of head that are used to produce orders that have a low quantity of products, are discs, or it is a special type of brush.

Figure 2-1: Drilling holes and punching brush material in the product

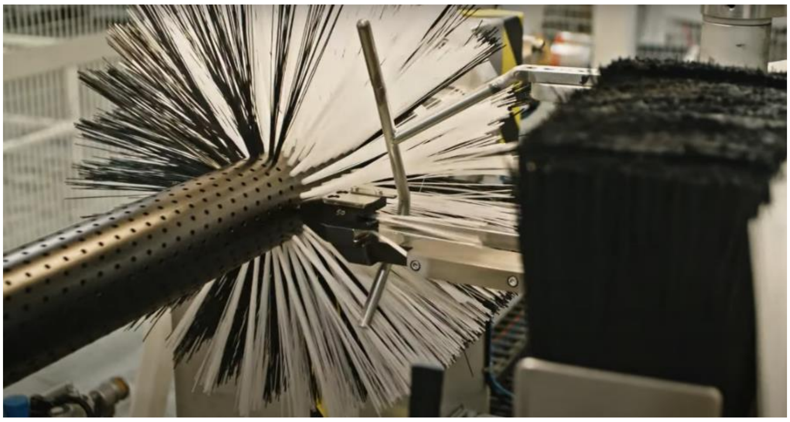

Figure 2-2: punching of brushes into the material

When necessary, the brushes need to be cut. This can be done on 1 of the 4 cutting machines. This needs to be done so the brushes have the same length or a pattern is needed in the brush. This is mostly done on the cutting machine, but can also be done by hand. Whether it is done manually or by machine depends on what is faster. For a roller, it is more efficient to put it on the cutting machine because it has more sides. The cutting machine can spin the roller alongside a sharp edge.

As seen in *Table 1,* the machines do not have the same production speeds, and not all types of products can be produced on every machine. The difference in performance is the effect of several things. Some machines are acquired later, the older machines have a worse performance due to wear and tear. This leads to breakdowns and more idle time for the older machines. The size of the machines also influences the performance. When a machine is bigger it can produce larger products. Since the machine is bigger, the distance between the product and the punching arm is bigger, increasing the time it takes to punch the brush into the material. Bigger products mean larger brushes and more distance between the holes. This leads to more time in between punches, so a lower speed. Lastly, products can have special patterns. This causes more calibration and getting the raw material in the right position, which leads to lower performance. Cutting time is not considered separately because that process is done at the same time the drilling and punching machine is running.

Table 1: Machine Characteristics

	Speed(holes per minute)	Type of product	Size?	Heads
Machine 1	100	Lats and rollers	Medium	5/7/10
Machine 2	200	Rollers	Small	3/5
Machine 3	100	Rollers	Small	3/5/7
Machine 4	80	Special rollers	Small	3/5
Machine 5	80	Big rollers	Big	5/7/10

2.3 Products

ABI can produce many different products with different materials, sizes, and patterns. The focus of this research lies in scheduling the production of the lats and rollers. There are also discs but they are produced on manual punching machines and that is outside the scope of this research. As a reference, the different products are depicted in *Figure 1-1 in Section 1.1*. The material in which the brush is put is referred to as the "raw material", the material of the rollers and lats. The end product can differ with regard to the length, type of raw material used, brush material, and, length of the brushes. These characteristics also determine on which machine the products are made. There are quite a lot of combinations regarding the raw material used and the brush material. The product can be of 5 different raw materials and the brush material of 8 different types. What material is used depends on the intended use of the product. The amount of work and time needed per product also varies. This is due to the varying number of holes that need to be filled and the size of the product. Because of all the variety in different types of products, the lats are never in stock, so they need to be ordered from a supplier. It can take a couple of weeks before the lats arrive and the production can start. In addition, some preprocessing is outsourced, when specific holes need to be made in the material, like threaded holes, or to saw the product to the right dimensions.

The rollers are kept in stock, but there is no inventory management system that forecasts the demand for the rollers. This means, when a big order comes in, there may not be enough rollers in stock. The lats are only produced on machine 1, this is the most efficient machine for this type of product, the rollers can be produced on all machines. The best machine for a certain roller depends on different variables of the roller, like the width of the holes or the dimensions of the product. More information about the machines is in *Section 2.2*. In conclusion, there are a lot of variables regarding the products that influence the production process of the products in the orders.

2.4 Production Process

The production process of ABI is described in *Figure 2-3.* As mentioned in S*ection 2.1*, there are 2 types of products are produced on the machines: rollers and lats. Rollers are kept in stock, but the lats always need to be ordered from a supplier. When the material is delivered, the preprocessing can start. The preprocessing is already done for the lats at the supplier, but for the rollers, it is done in the workplace. Preprocessing includes drilling holes on the surface that cannot be done with the machine, like drilling a hole through the middle of the roller. Also, the products need to be sawed to the right dimensions.

Then, the products go on the drilling and punching machine. After the raw material is filled with brushes it is manually removed from the machine and some products need to be finished by cutting the brushes to the same length. This can be done by hand, but also by machine, depending on the product. Depending on the material and what the margins are, the material needs to be sawed again, because the material can expand after the brushes are punched in. For some orders, after the brushes are put in the material, screw thread or holes need to be put into the material. As seen in *Figure 2-1,* finishing includes the actions of cutting, sawing, and, drilling.

Some orders need to be made by hand. This is necessary for the steel brush material, the discs, when the brush material needs to be put in at a different angle than 90 degrees, or when the quantity is too low, so it is more efficient to produce manually. There can be a difference in hole diameter, brush material or length, or pattern, for each order. These variables need to be set up in the drilling and punching machine, which takes time.

When the order is produced it needs to be packaged. This process varies per product type. The small rollers go straight from the machine into a cardboard box on a pallet. That pallet can be easily transported to the packaging station where it will be covered in bubble wrap and plastic. From here the order is moved to the loading part of the production hall.

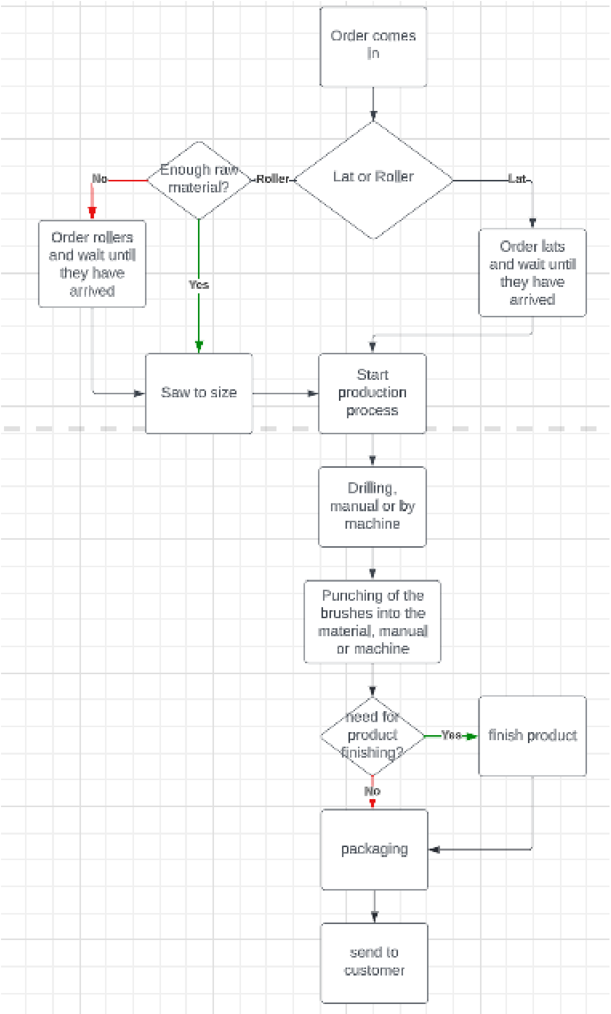

Figure 2-3: the production process of ABI

2.5 Scheduling Process

To give insight into the current scheduling process of the orders on machines at ABI, the actions needed for an incoming order are described. A customer calls and wants to order a specific type of technical brush. This can range from a small batch of 1 or 2 products to a large batch of 1000 in the order. The kind of material and specifics of the brush will be determined by the needs of the customer. The intended use of the brush will decide variables like the width of the hole in the material, the brush length and material, and the pattern in which the brushes are punched. A price is discussed and the order is finalized and put in the ERP system with an order number. The standard delivery time is set at 3 weeks after the order is put in the ERP system and communicated to the customer.

Following this, the order is checked whether any materials need to be ordered. On Friday, the week before the due date, the order is put on the overview list along with all of the remaining orders of previous weeks and all orders that are scheduled to be produced in that following week. On the overview list, it is stated in which week every order needed to be produced, so the backorders are produced first. If production can keep up with the inflow of orders, the overview list would function as a list of orders that need to be produced in the following week. However, in the current situation, production cannot keep up. This means the overview list grows every week with backorders and incoming orders. This way, there are always backorders since the backorders will be produced first, leaving no time for the remaining orders. Only if there are fewer incoming orders, the backorders will reduce.

The production sequence of the orders is determined by the Chief of production. He looks at the overview and decides what order is going to be produced. In his decision-making, the due date can be overlooked and some other order, which has come in more recently, is produced. This can happen because a customer requests to fast-track their order. This does not happen often, and only against compensation or regular customers can do this. Another reason that orders are surpassed is because it can take less effort or time to switch to an order that arrived later, because of setup times. A third reason an order can be overlooked is if the quantity of products in the order is low since a large order can be produced more efficiently. This leads to more delays for the remaining and delayed orders. Since there is no schedule to fall back to, there is no certainty when a specific order is going to be produced. The result is that there is no concrete information for the customer about the delivery time of their order, only when their order has started being produced. That is why the standard delivery time is set at 3 weeks, but this is an estimation based on previous years of production and ABI cannot make that due date anymore.

2.6 Currently Available Data

The data available is information on the number of orders, how many products are in that order, and which machine is used for the order. This data about the orders is recorded in the production overview. The overview is updated every week, produced orders are removed from the list and new orders are added. The updated overview list is the new overview list and the previous version is thrown away. For this research, the overview lists are kept and used for analysis. However, this data is only updated every week. So, information about the backorders and due dates of orders are only known for that week. This means there is no recorded daily information about what order is being produced. Therefore, the sequence of the orders that are produced and the number of fast-tracked orders is unknown. What we can use from the overview list, is information about the number of orders with the number of products that need to be produced and which machine is used. Based on this information the percentage of the number of orders and the number of products that are not delivered on time is calculated. Also, the number of orders that need to be produced per machine is known. This data is located in *Appendix A.* For every overview list available we calculate the percentage of backorders and the percentage of the products that are in those backorders. These different percentages are needed because the number of products in an order can vary a lot. Based on the data from all of the available overview lists an average of 68% of total products and an average of 60% of total orders that are on the overview list are backorders. This shows that ABI has a lot of backorders and needs a production schedule since it gives ABI more insight into the realistic delivery time of their products. Then, this information can be shared with the customer providing more clarity about the status of their order.

For every order important information is needed to make a production schedule like, the number of holes in the product, when the order is placed, and when the ordered raw material comes in. ABI does not log the inventory of the rollers and the brush material; therefore, there is no available information. The inventory for the rollers is kept through observation of the chief of production. There is no estimated demand, so this reorder point is an estimation of the demand for the rollers. However, this is an educated guess by the chief of production. This way of reordering is fine for now but will be a problem when he stops working and the knowledge leaves with him.

2.7 Conclusion

In this section, we answer the research question:

- What is the current situation of ABI, regarding the production process and the production schedule?

This chapter is mostly important to gather information about the scheduling situation of ABI. The current situation of ABI can best be described in *Figure 2-3.* This figure depicts the entire production process a product has to go through to be completed. The type of products that need to be scheduled are the rollers and the lats. Based on the characteristics of these products they need to be produced on different machines. The different kinds of machines are the lathe, the drilling and punching machines, and, the cutting machines. After the products are completed, they need to be sent to the customer.

The scheduling process is manual and can be described in a few steps:

1. An order comes in and is entered into the ERP system
2. All of the materials needed for producing the products are checked or ordered.
3. The order is put on an overview list ready to be produced on the drilling and punching machine.
4. The Chief of production looks at the overview list and determines which order will be produced next.

3. Literature Study

We consult the literature in this chapter to gather information regarding different machine scheduling situations and their characteristics. These machine-scheduling situations are described in *Section 3.1.* In *Section 3.2,* we explain how scheduling works and what characteristics are important. In *Section 3.3*, we investigate different scheduling methods that can be used to make a schedule, together with methods to increase the efficiency of a schedule. We conclude this chapter in *Section 3.4,* with an answer to the research question mentioned in *Section 1.4*. In Chapter 2, we talked about products going on machines to be produced. However, in the literature, the terminology for producing a product with a machine is processing a job through a stage.

3.1 Machine Scheduling

There are different machine environments with their classifications and characteristics. If a job only has to go through 1 machine it is a single-stage job, but when it has to go through more than one machine before the job is fully processed, it is considered a multiple-stage job. Within machine environments, there are differences in the way of scheduling. However, all the environments have a similar situation, there are a number of jobs that need to be scheduled on a number of machines.

3.1.1 Single-Stage Job Scheduling

Single-stage job scheduling means that there are a number of jobs that need to be scheduled on a machine. Within single-stage job scheduling, there are different types called single-machine scheduling, parallel-machine scheduling, uniform-machine scheduling, and unrelated-machine scheduling. Uniform-, and unrelated machine scheduling can be seen as a form of parallel-machine scheduling. For all the environments, the objective function measures the quality of the schedule and determines the sequence in which the jobs are scheduled, more information about objective functions is discussed in *Section 3.2.1.*

3.1.1.1 Single-Machine Scheduling

In this situation, there are a number of jobs that need to be processed on 1 machine. Since there is only one machine, the sequence in which the jobs are scheduled is important because as opposed to the other environments there is only 1 machine. This means that there are fewer possible sequences. In the most simple situations with this problem, there are no setup times or due dates, just the processing times of a job. However, to resemble the real-life situation better, variables like release dates, setup times, and, due dates can be considered when scheduling the single machine. (Senthilkumar & Narayanan, 2010)

3.1.1.2 Parallel-Machine Scheduling

In parallel-machine scheduling, a job has multiple machines on which it can be scheduled. Each job needs to be assigned to one of these machines. With parallel-machine scheduling the characteristics are identical for each machine. This means the processing times of a job are the same for every machine. When a setup time occurs, multiple identical machines can help optimize the makespan by clustering the same kind of jobs together. (Cheng & Sin, 1990)

3.1.1.3 Uniform-Machine Scheduling

Uniform-machine scheduling has the following characteristics. The job can be processed on all the machines. However, contrary to parallel-machine scheduling, the machines have different processing speeds for the jobs. This type of machine scheduling is more difficult than parallel-machine scheduling since the process times are different per machine. This means a decision needs to be made about what job is processed on what machine to improve the efficiency of the schedule. When producing as efficiently as possible all jobs are scheduled on the fastest machine. This has a consequence that the makespan will be bigger, as opposed to when the slower machines are also used. So, a trade-off needs to be made between the jobs, go through the faster machine or the slower machine. (Senthilkumar & Narayanan, 2010)

3.1.1.4 Unrelated-Machine Scheduling

When there is a difference between the processing speeds of jobs on machines it is called unrelated-machine scheduling. So, it depends per job what the processing speed is per machine. This means some jobs are more efficient to process on a certain machine. The reason can be due to technological differences in the machines or the features of the jobs. (Senthilkumar & Narayanan, 2010)

3.1.2 Multi-Stage Job Scheduling

The scheduling situation for multi-stage job scheduling problems is the same as for single-stage scheduling problems. However, with multi-stage scheduling, the jobs have to go through multiple stages or multiple (different) machines. There are 3 different kinds of scheduling characteristics and problems. Depending on the type of the jobs the scheduling problems can be classified into *flow shop*, *job shop,* and *open shop*. With flow shop scheduling all jobs go through the machines in a fixed order. With job shop scheduling the sequence in which the jobs go through the machines, the number of machines, and which machines the jobs need to go through are arbitrary, but known in advance. With open shop scheduling the sequence of machines the jobs have to go through does not matter and is arbitrary and not known in advance. Lee & Yamakawa (1998)

3.1.2.1 Flow-Shop Scheduling

In flow-shop scheduling, each job is processed at 1 machine in each stage. The jobs that need to be processed all need to follow the same operational order or processing route. A job consists of multiple operations and needs to go through multiple machines. The first operation needs to be processed on machine 1 and the second operation on machine 2, and so forth. Regarding this scheduling type, the processing time at each stage for each job is known in advance. The difficulty with this type of scheduling is the difference in processing times between each machine which can lead to a lot of idle time for a machine. This difference in processing time between machines can lead to an overall poor performance. (Naderi, Zandieh, Khaleghi Ghoshe Balagh, & Roshanaei, 2009)

3.1.2.2 Job-Shop Scheduling

Just like flow-shop scheduling, the jobs have a certain processing route. However, the difference from flow-shop scheduling is that the route of the jobs can be different from job to job. Because the sequence of the machines used per job can be different it is very difficult to schedule the jobs efficiently, especially when the number of jobs that need to be scheduled increases. It can be the case the majority of the jobs need to pass a certain stage, creating a queue at that stage. This causes the jobs to take longer to be completed, which is not desired. This problem is largely applicable to job-shop scheduling because the number and stages that need to be completed are arbitrary per job. (Balas, Simonetti, & Vazacopoulos, 2008)

3.1.2.3 Open-Shop Scheduling

In open-shop scheduling, there is no process sequence for each job. This means that for the jobs it does not matter whether it first goes through machine 1, or machine 2. It is also possible a job does not even need to go through a certain machine. The freedom in the sequence of the machines that the jobs need to follow can lead to the schedule becoming very efficient and without queues. However, the freedom in the sequence of the machines the jobs need to follow causes a large number of possibilities, leading to a high computation time. (Senthilkumar & Narayanan, 2010)

3.2 Scheduling

When a job is scheduled carelessly, it can lead to an inefficient schedule. By allocating resources to activities in a specific way the organization can maximize the utilization of these resources and increase productivity. This can be done via different scheduling methods, discussed in *Section 3.3*. In the scheduling methods, an objective is chosen that dictates the way the jobs are scheduled. Some of these objectives are to minimize the maximum makespan, minimize maximum lateness, and shortest processing time. (Balas, Simonetti, & Vazacopoulos, 2008).

Machine scheduling problems are optimization problems. With machine scheduling problems there can be a lot of different schedules, the problem is finding the best schedule or solution. In machine scheduling problems, the problem situation can be expressed in an optimization model that represents the real situation. This real situation is simulated with an *objective function, constraints,* and *assumptions*. (Balas, Simonetti, & Vazacopoulos, 2008). An example of such an assumption is that a machine is always operational. Of course, this is not the case in a real situation but implementing this in a model is difficult, because of the uncertainty of the machine breaking down. Constraints are the setup times between jobs, release dates, due dates, and uncertainties like machine breakdown. The optimization model needs to take these constraints into account to resemble the real situation. However, when an optimization model needs to consider a lot of constraints the computation times to find the best solution will be very high. Also, with every additional job that needs to be scheduled, it takes exponentially more time to compute an *exact solution*. An exact solution is the best possible solution there can be resulting from the optimization model. Therefore, a trade-off can be made between the time it takes to generate a solution and the quality of the solution. (Sanlaville & Schmidt, 1998).

3.2.1 Objective Functions

As mentioned in *Section 3.1.1.1,* an objective can be used to schedule the jobs. Possible objectives for a schedule are to minimize the *makespan*, minimize the *tardiness*(lateness), and minimize the *machine's idle time*. Makespan is the total length of the schedule. Tardiness measures the delay of the jobs that are scheduled, a delay occurs when the completion time of a job exceeds the due date. Machine idle time is the time the machine is not processing jobs, this can occur when the machine needs to be set up before an order can be processed. Minimizing the makespan can be used to generate an efficient schedule. Minimizing tardiness can be used to make sure customers get their products on time. Minimizing the idle time makes sure the machine is processing as efficiently as possible. When an objective is chosen for a schedule all of the jobs are scheduled so that objective is met. A downside is that other objectives are overlooked. For example, when the objective is to minimize the machine's idle time in a schedule, it could be the case that there are jobs with high tardiness. (Liaee & Emmons, 1997) For this reason, multiple objectives can be obtained to solve scheduling problems. This means when the jobs are scheduled on the machines more than 1 objective is taken into account. However, having a strategy to optimize multiple objectives becomes very difficult to solve and takes a lot of time to compute. (FAN, 2020)

A job scheduling problem gets increasingly difficult when multiple variables or constraints have to be taken into account, especially when variables are dependent on each other. For example, it becomes more difficult to schedule jobs if the setup times between different jobs are different. This difference in setup times between jobs on the machines occurs when the jobs are in a different *family*. A family in this context is a set of jobs that have the same characteristics so there is no setup time before the jobs can be processed on a machine. So, when 2 jobs are scheduled after each other that are in the same family there is no setup time, but when 2 jobs from a different family are scheduled after each other there is. Additionally, there are *sequence-dependent* and *sequence-independent setup times*. with sequence-dependent setup times, the setup time depends on the preceding job and the current job. With sequence-independent setup times, there is a setup time that only depends on the current job. (Liaee & Emmons, 1997).

3.2.2 Job Characteristics

The job characteristics determine together with the objective of the schedule in what sequence the jobs are going to be scheduled. Important characteristics are the *release* and *due dates*, and the *processing time*. The release date is used to decide after which date the job can be scheduled. A reason a job has a release date is that not all of the material to complete the job is available before a certain time. The due date is the date by which the job should be completed. When the job is completed before the due date it can be considered early, when it is completed after the due date it can be considered late. (Balas, Simonetti, & Vazacopoulos, 2008)

Jobs can be *non-preemptive* or *preemptive*. It is called a non-preemptive job when a job is completely processed on a machine without interruption. It is called a preemptive job when it can be discontinued before its completion time and reassigned to either the same machine or some other machine and continue processing. (Cheng & Sin, 1990)

3.2.3 Online and Offline Scheduling

A distinction can be made between offline and online scheduling. With offline scheduling, the release date, processing time, due date, and other necessary data of each of the jobs are known before determining the schedule of the jobs. With online scheduling, algorithms make scheduling decisions at each time instant when the available jobs are scheduled, based upon the characteristics of the jobs that have arrived thus far without knowledge of jobs that may arrive in the future. Online scheduling can be classified into different types, according to Senthilkumar & Narayanan (2010).

- *Scheduling jobs one by one:* The available jobs are ordered in a list, and the objective of the schedule determines the sequence. Then, each of the jobs will be assigned one by one to a machine, if there are multiple. This happens before the next jobs that need to be scheduled come in.
- *Unknown running time:* The processing times of the jobs are unknown until the job is completed. The online scheduling algorithm only knows whether a job is still processing or not. The sequence is determined in the same manner as by scheduling the jobs one by one.
- *Jobs arrive over time:* The processing time of each job is known at the time of arrival of that job. However, the arrival time of each job is not known in advance.
- *Interval Scheduling:* Each job needs to be executed in a predetermined time interval. The job will be rejected if it is impossible to execute it in that time interval. The objective is to maximize the number of accepted jobs. (Senthilkumar & Narayanan, 2010).

3.3 Scheduling Methods

To solve the machine scheduling problems discussed in *Section 3.1* there are numerous scheduling methods. These scheduling methods are used to determine the sequence of the jobs that need to be processed on a machine. This can be done with an exact solution algorithm that seeks the optimal solution with regard to the objective function. For most machine scheduling problems, this can only be done when the number of jobs that need to be scheduled is low.

When the number of variables like job characteristics, the number of stages a job has to complete, or the number of jobs that need to be scheduled increases, it becomes more difficult to find the best schedule. The computation time increases because more variables need to be considered to find the best solution. If the variables that need to be considered increase too much an exact solution cannot be found within a reasonable time. To work around this problem heuristics are used. These methods provide a feasible solution that tries to achieve the objective of the schedule while keeping a small computation time. This is not necessarily the optimal solution, but finds a solution to the scheduling problem.

3.3.1 Priority-Rule-Based Scheduling

Priority-rule-based scheduling consists of two parts: the schedule generation scheme and the priority rule. A generation scheme provides the way how a schedule is constructed. A job can have multiple operations, and before the next job can start on a machine the current job needs to be completed. Two schedule generation schemes can be distinguished, the serial and parallel schemes. Both schemes construct a schedule by repeatedly extending a partial schedule until a complete schedule is obtained. The difference in generation schemes is that a serial generation scheme makes use of an *activity-incrementation* scheme, while a parallel generation scheme follows a *time-incrementation* scheme. In both approaches the job is scheduled with the highest priority. Activity-incrementation approach entails a decision being made about which job is scheduled, then the time in the schedule increases to the completion time of that job whereafter the next job is scheduled. Time-incrementation approach entails that at a certain time decisions are made to schedule the jobs that are currently available to be scheduled on the available machines. The next decision is to schedule what job on which machine is made when the job is completed. So, with serial generation schemes, there are as many scheduling decisions as jobs, whereas with parallel generation schemes the number of scheduling decisions is not known beforehand. (Vanhoucke)

With serial schedule generation schemes, there are a number of stages equal to the number of activities, or jobs. There are three kinds of sets, the remaining set, the decision set, and the completed set. The remaining set describes all the jobs that still need to be scheduled. The decision set contains jobs that are not yet scheduled and whose predecessors are in the completed set. A predecessor is a job that was in the decision set but is now in the completed set. The completed set consists of all the jobs that have already been scheduled. The schedule is made in the following way. First, a job is chosen from the decision set with the highest priority value. This job is scheduled as early as possible and moves to the completed set. This repeats itself until all the jobs are scheduled.

With parallel schedule generation schemes there are a number of stages N, where either no, one, or multiple activities are scheduled. There are four sets of activities, different from the serial schedule generation schemes. The completed set consists of jobs that are scheduled and completed at the end of stage n. The active set contains all jobs that are scheduled, but not yet completed at that time in stage n. The decision set contains all unscheduled jobs that could be scheduled at that time in stage n. Lastly, the remaining set contains all jobs that are not yet scheduled and that are not available for scheduling at that time in stage n.

To determine the sequence of the jobs within a schedule generation scheme, a priority rule can be imposed. A priority rule uses a function to give a priority value to each activity in the decision set. All jobs in the decision set are reviewed and the job with the highest priority value is selected to be scheduled. Characteristics of priority rules are that they require little computation time, and are intuitive, robust, and easy to implement. (Vanhoucke)

A distinction can be made between local and global priority rules. Local rules only use information about the current operation and the machine on which this operation is processed. Global rules use more information, about the entire process a job has to follow and about other operations or machines the job has to complete.

Regarding priority rules, a distinction can be made between dynamic and static rules. Dynamic rules return a priority value for a job depending on the stage it is performed in, while static rules return the priority value regardless of the stage. With static rules, the priority value of an operation does not change during the execution of the priority rule, while the priority value changes over time with dynamic rules. There are a lot of different priority rules, some examples that are described in the literature are earliest due date, first-in first-out, shortest processing time, maximize minimum lateness, and minimize total makespan. (Schutten, 1996) (de Boer, 1998)

3.3.2 Metaheuristics Algorithms

A heuristic is a technique aimed to solve a problem faster when other methods that search an exact solution are too slow or difficult to compute. A metaheuristic is a higher-level technique or heuristic that seeks, generates, or selects a heuristic that may provide a sufficiently good solution to an optimization problem (Attea, et al., 2021). These newer techniques were developed to address problems such as nonlinearity, multi-objective, and uncertainty. (Fayaed, El-Shafie, & Jaafar, 2013). These algorithms attempt to find the best viable solution out of all solutions to an optimization problem. Potential solutions are evaluated and the algorithm performs a series of operations to find a different, better solution. Due to the scope of this research, we will only discuss the local search metaheuristics. (Glover & Sorensen, 2015).

Local search algorithms start with a feasible solution, for instance, provided by a priority rule, and try to improve that solution. The solution is improved by making small changes to that first feasible solution. For instance, trying to achieve a better schedule by changing two jobs with each other. If the schedule is improved upon the chosen objective, the schedule is considered better. This new schedule is called a neighbor solution from the current solution, all the neighbors together from a solution are called the neighborhood solutions. A move can be made to a neighbor solution to improve the objective of the schedule. In the literature, many methods are discussed to improve a solution. However, Steepest Descent, taboo search, and simulated annealing are most frequently mentioned when we look into the literature. The first is a fairly simple search strategy, it considers the neighborhood solutions and chooses the best solution if it is better than the current solution. Then, the neighborhood solutions of the new solution are considered and again the best neighbor solution is chosen as the new solution. This process repeats itself until the solution cannot be further improved, this solution is a local optimum. Often, this results in poor-quality solutions. The quality can be increased to run the algorithm multiple times with different starting solutions.

Another local search strategy is the taboo search algorithm. With this approach, the iteration can move to a worse solution, if no better one exists in the neighborhood of the current solution. A taboo list is kept that has a short-term memory of the last few moves to avoid jumping from one solution to the other and back again. The taboo search can have an attempt limit to avoid the algorithm to keep exploring all options or a specific threshold that is used for the increase in the objective of the solution.

Simulated annealing is a local search algorithm that picks a random neighbor of a current solution. When a neighbor solution is better it becomes the current solution. When the neighbor solution is worse than the current solution it still has a probability of becoming the current solution. This probability depends on the difference between the value of the current solution and the neighbor, and a control parameter. This control parameter decreases during the execution of the algorithm making it less likely to move to a worse neighbor. Eventually, the control parameter makes sure no worse solution is picked. (Schutten, 1996) (Glover & Sorensen, 2015)

3.4 Conclusion

To conclude this chapter we answer the research question:

- What kind of machine scheduling problems situations are discussed in the literature, and how is a schedule made in those situations?

There are several different scheduling situations:

- Single-stage job scheduling, There is only 1 stage(machine) a job(product) has to go through to be completed, the different single-stage scheduling situations consists of:
- Single-machine scheduling: there is only 1 machine the jobs can be processed on
- Parallel-machine scheduling: there are multiple identical machines on which a job can be processed
- Uniform-machine scheduling: there are multiple machines with different processing speeds on which a job can be processed
- Unrelated-machine scheduling: there are multiple machines with different processing speeds determined by the job due to technological differences between the machines
- Multi-stage job scheduling, before a job can be completed it first needs to go through a number of stages, the multi-stage job scheduling situations consist of:
- Flow-shop scheduling: each job follows the same, previously known, sequence of stages before it is completed
- Job-shop scheduling: each job can follow, a previously known, different sequence of stages before it is completed
- Open-shop scheduling: each job does not have a process sequence and per job, it also differs what path they follow, which is not previously known

Jobs need to be scheduled a certain way in these scheduling situations to be produced as efficiently as possible. There are different methods to schedule all of the jobs. To find an exact answer is very difficult and quickly leads to an unsolvable problem. For this reason, we can use heuristics and priority rules to find a feasible schedule. Priority rule-based scheduling consists of 2 parts: a schedule generation scheme and the priority rule. There is a serial and a parallel generation scheme that works as follows. A serial generation scheme uses activity incrementation, and a parallel generation scheme uses time incrementation. The priority value is calculated for every job, the job with the highest priority is scheduled. This process repeats itself until all of the jobs are scheduled. An example of a priority rule is the earliest due date, which tries to minimize the maximum lateness of a schedule. Priority rules do not create the most optimal schedules, so to improve the initial schedule a (meta)heuristic can be used, for example, Steepest Descent, taboo search, and, simulated annealing. These methods work differently from each other, but essentially the methods check alternative schedules and try to find a better solution.

4. The Scheduling Problem and Solution Method

In this chapter, we use information from *Chapter 2* and *Chapter 3* to model the situation of ABI. This chapter provides a solution that can be implemented in their scheduling process. In *Section 4.1,* we discuss the situation of ABI to have a good overview of what part of the production process we model. In *Section 4.2,* we explain and discuss the model of the situation described in *Section 4.1*. In *Section 4.3,* we choose and explain the method to schedule the orders. Also, we explain the method we implement to improve on the initial schedule. In *Section 4.4*, we answer the research question related to this chapter.

4.1 Situation of ABI

The production process at ABI can be described as multi-staged. In *Figure 4-1,* the stages an order has to complete are depicted. There are multiple stages an order has to go through to be completed. However, not all stages need to be completed for all orders, *Figure 4-1* shows possible different paths. For clarification, the black line goes through all stages, whereas the colored paths are the possible paths an order can follow, deviating from the black line. The rollers have to pass a stage the lats do not need to go through, this stage is called preprocessing. During preprocessing, specific holes can be made in the rollers and they need to be sawed to size. Both types of products have to go through the drilling and punching machine. Then, most products need finishing, and finally, the products need to be packaged. So, 4 is the maximum number of stages an order has to go through and 2 is the minimum. As depicted in *Figure 4-1,* the products follow a path from left to right, meaning a product cannot go from stage 3, finishing, to stage 2, drilling and punching.

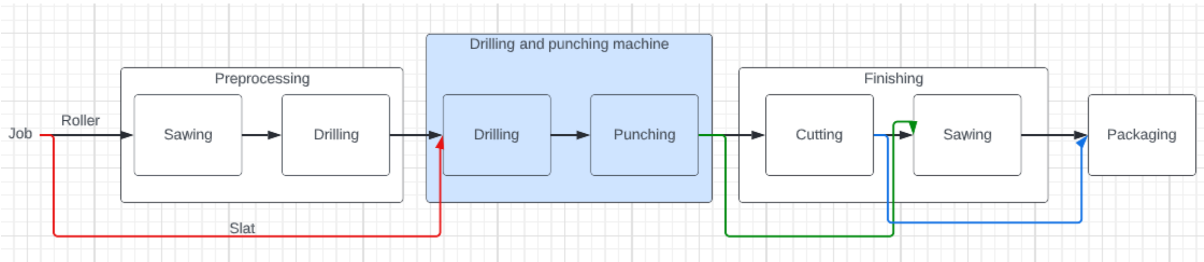

Figure 4-1: production path of an order

The situation of ABI is described in *Chapter 2*. So, here a quick overview of the products, machines, and production process of ABI is given. This information is important to classify the scheduling problem and find the best method to handle this problem.

- Lats
  - Are purchased from a supplier and ready to be processed
  - Not in stock
- Rollers
  - Are in stock
  - Require preprocessing
- Preprocessing
  - This is done manually
  - Preprocessing consists of sawing the rollers to size, and drilling specific holes the drilling and punching machine cannot make
  - After preprocessing the rollers are stored in a buffer before processing on the drilling and punching machine
- Drilling and punching machines
  - There are 5 different drilling and punching machines
  - 2 activities are completed on the machine, drilling holes in the material and putting brush material in those holes
  - All orders need to go through this stage

- Finishing
    - Not all orders need to go through this stage, it is possible the products in the order only need to be cut, sawed, or no finishing is required
- Packaging
    - All products need to be packaged

4.1.1 Current Machine Scheduling Method

In the current situation, the orders are not scheduled. There is no production schedule for the orders that need to be processed. There is only an overview of what order should be produced on what machine and in which week that should be. This is a problem because it leads to uncertainty about when an order is processed. This uncertainty leads to unavailability of the information for the customer about the delivery time of their order. Based on the size of the hole that is made in the material of the products, the orders are put in an *order family*. When an order from a different order family has to be processed, the drilling head needs to be changed. This change of heads takes 30 minutes. The setup times are *sequence-independent*, it does not matter whether the heads are changed from 5mm to 3mm, or 7mm to 3mm. There is also a setup time between each order of 30 minutes to adjust the machine to the dimensions of the products.

What machine is used for each order is chosen by the office team in discussion with the head of production. Most products can only be processed on a certain machine. However, for some orders, the experience is needed to allocate them to a faster or slower machine. This allocation depends on the urgency of the order, the setup-related order families, and how many orders need to be scheduled. If there is no need to use all of the machines, only the most efficient machines are used. This can occur when there are no backorders, otherwise, all of the machines are in use. There are quite some variables that need to be taken into consideration when allocating an order to a machine. Some examples of these variables are the size of the product, size of the brush material, kind of brush material, number of products, and what kind of product it is. For this research, there are too many variables to consider and a specific machine is chosen by the employees as it is done currently.

4.2 Model of the Scheduling Problem of ABI

We consider the situation described in *Section 4.1* and use the theory in *Chapter 3* to classify the scheduling problem of ABI and model the situation. The model is described in *Section 4.2.2.* This model tries to reflect the real situation of ABI. There are several reasons for using a model. The real situation has too many variables that are dependent on each other to be able to create a schedule. We use assumptions to simplify reality and to model the situation. Constraints are used to model characteristics of the orders, so a feasible solution can be created. Additionally, a model of the situation can be used to compare it to similar situations described in the literature. Then, this model needs to be translated into the operation of ABI.

4.2.1 Drilling and Punching Stage

The only part that is modeled is the drilling and punching stage. This is the stage that is modeled, and that has a few reasons. The main reason is that the drilling and punching machines are the bottleneck of the production process. If a schedule can be made for the drilling and punching stage, the entire operation can use that schedule. The orders can be preprocessed by the same schedule and the orders are always ready because this stage has a lower processing time. If the products in an order need finishing it can be done while products of the same order are still on the drilling and punching machine. In *Figure 4*-2, a *transfer batch* is depicted. A transfer batch is when the products of an order can be processed on the next machine while products of that same order are still being processed on the previous machine. As can be seen in *Figure 4-2,* the total makespan of the order does not increase much when the products need to go through the finishing stage. (Schutten, 1996). The products in the figure are from the same order.

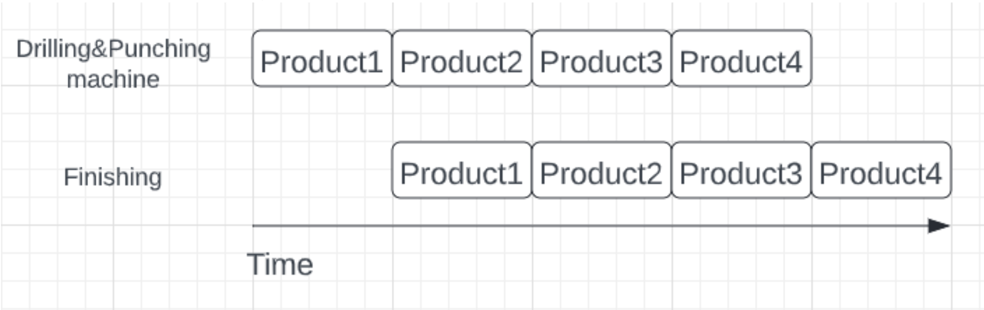

Figure 4-2: view of the efficiency of a transfer batch

The drilling and punching stage is modeled is because it is the only stage with setup times. This means the sequence of the orders in this stage is more important than in the other stages because it relates to the efficiency of the production process. Also, all products need to go through this stage, which is not the case for all stages. Lastly, the drilling and punching stage is the best stage to model because the orders are not processed manually. So, the processing times can be estimated.

4.2.2 The Model

We have made assumptions and constraints for the model, to reflect the real situation. To model the real situation as best as we can, we make estimations. The values of the processing speeds are estimated but can be adjusted if it seems that after modeling the orders the processing times are not accurate. By adjusting variables the model can be more accurate. ABI has multiple drilling and punching machines, but the model describes only machine 1. The objective of the model is to schedule the orders in an efficient way using priority rules, this is explained in *Section 4.3.1.*

Assumptions

- The setup time between orders is 30 minutes to adjust the dimensions of the products
- The setup time between orders with a different size head (hole) takes 30 minutes on top of the initial setup time
- The processing time is 100 holes per minute
- When an order is scheduled to be processed on the machines all preprocessing is ready
- The machines can work 8 hours a day, uninterrupted and there are no breakdowns
- Drilling the holes in the material, after which the brush material is punched in is seen as one operation
- The orders are non-preemptive, this means an order is completely processed without interruptions
- The values for the release dates and due dates are integer values

Constraints

- Orders with release dates cannot be scheduled before that time
- The machine can only process 1 order at a time

The input for the model will be 5 data sets with each having around 15 orders. The data sets contain information that is needed for scheduling the orders. The first is information ABI needs to distinguish the order, like customer name, number of the order, and what kind of product it is. Secondly, information is needed to calculate the processing time of the order. This contains the number of holes in the products and the number of products in the order. Third, the type of head needed for that order is needed for the setup times, since switching heads takes another 30 minutes. Fourth, the information for the release date of the order is needed. This relates to the supplier delivery time of the material for the lats. This can also be used if the rollers cannot be processed before a certain time. For instance, when there is not enough raw material. Lastly, the due date is needed for the priority of the orders. All of this information per order is stored in a list. The way the schedule is determined will be described in *Section 4.3.*

In the model, an integer value is given to the release date and the due date. This is done so it is easier to calculate the priority of the orders and when they can be scheduled. The dates are made integer values by assigning a number to a date, for instance, 01-01-2023 corresponds with the number 1. After the sequence of the orders is determined the translation to reality needs to be made again. In the MS Excel model, this translation can be made by transferring the orders to another sheet into a Gantt chart that displays the number of hours an order is being processed on the machine. A day consists of 8 hours. The orders are non-preemptive, as mentioned in the assumptions.

4.3 Scheduling Method

ABI wants to have more insight into their production process and when the orders are ready to be delivered, to communicate this to the customer. To do this, a schedule needs to be made. For the schedule, the release dates and the due dates must be taken into account. In *Section 4.2* the modeled situation is described. Now, a method to schedule the orders needs to be chosen. A priority rule can be used to accomplish numerous objectives for the schedule, e.g., minimum makespan or minimize the maximum lateness. The makespan is the total time it takes for all of the orders to be done processing. Minimizing the maximum lateness is to make sure the lateness in the schedule is kept at a minimum.

In this situation, there is a single-machine scheduling problem with release dates, processing times, family setup times, and due dates. The way the orders are scheduled is *offline*. This means all of the data needed to schedule the orders is known in advance, including information about the release date, processing time, and due date.

4.3.1 Priority Rule

As explained in *Section 3.3.1,* priority-rule-based scheduling consists of 2 parts. The schedule generation scheme and the priority rule. The generation scheme and the priority rule are used to construct the schedule. There are 2 kinds of generation schemes, serial and parallel. Both schemes construct a schedule by repeatedly extending a partial schedule until a complete schedule is obtained. This is done by looking step-by-step for the order with the highest priority and scheduling it until all of the orders are scheduled. The difference between those 2 is that a serial scheme considers the priority value for all orders and proceeds to schedule the highest priority order. As a consequence, the serial can have more idle time. Because of the release date, there can be a lot of idle time in the schedule. With parallel scheduling, there is less idle time. This is because the decision set consists only of the orders that are available to be scheduled at the earliest possible time. From those orders that are in the decision set the priority value is calculated and the order with the highest priority is scheduled. For ABI the machines must have little idle time, to have the optimal schedule is less important than that. So, the orders are scheduled via the parallel method.

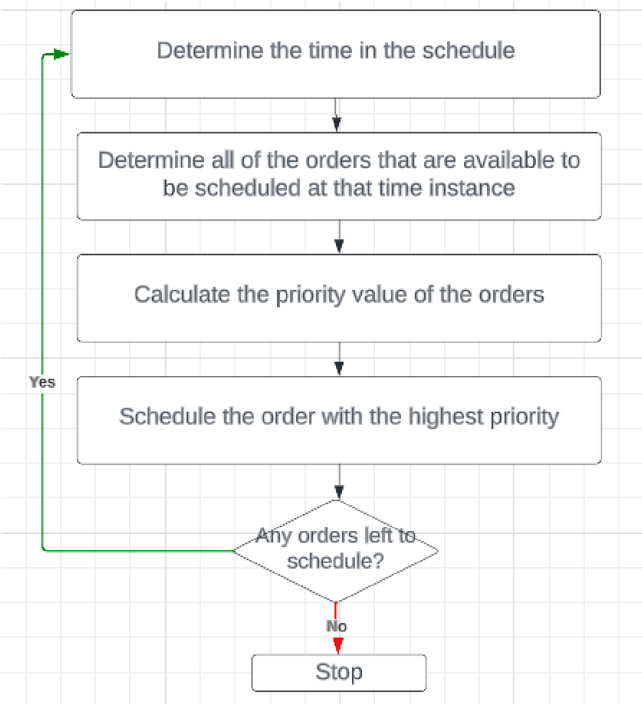

Figure 4-3: Flowchart of the priority rules

To explain and make the priority rules more clear we use letters, underneath the meanings are explained.

$$d_i = due\ date\ of\ order\ i$$
$$s_i = setup\ time\ of\ order\ i$$
$$c_i = completion\ time\ of\ order\ i$$
$$p_i = processing\ time\ of\ order\ i$$
$$EDD = Earliest\ Due\ Date$$
$$MinSlack = Minimum\ Slack$$
$$SST = Shortest\ Setup\ Time$$

For calculating the priority given to the orders priority rules are used. The rules most important for ABI are the Earliest Due Date(EDD), the minimize slack rule(MinSlack), or the Shortest Setup Time(SST). The priority per rule is calculated in the following way:

$EDD = c_i - d_i$, the lateness per order is calculated, the order that is the latest is scheduled.
$MinSlack = d_i - p_i - c_{i-1},\ i-1$ indicates that the completion time of the previous order is used. The order with the least amount of "*Slack*" has the highest priority to be scheduled. Slack is the time between when an order is completed and the due date. Slack can be negative, which means the order is completed past the due date.
$SST = s_i$, the lowest value of the setup time has the highest priority to be scheduled

ABI wants to have the same delivery time for all of their customers, so the due date is very important in determining the priority value of an order. An order that comes in last will have a higher due date and therefore a lower priority to be scheduled than an order that came in before. Also, the efficiency regarding setup times in the schedule is important for ABI. In practice these rules can be combined, the priority can depend on both the due date and take into account the setup time. How important the setup time is, can be determined by a weight given to the setup time. If the weight is high, the orders with setup times have a lower priority. A high value for the weight leads to orders being clustered by their setup family. In the case of ABI, this is the size of the heads. So, an order that has the same hole size in the product has a setup time equal to 0. The EDD and MinSlack priority values are then calculated in the following way:

$$EDD = c_i - d_i - s_i$$
$$MinSlack = d_i - p_i - c_{i-1} + s_i$$

4.3.2 Steepest Descent Method

As mentioned, parallel scheduling and priority rules most likely do not result in the optimal schedule. We use the Steepest Descent method to improve the schedule the priority rule gives. In *Figure 4-4*, a flowchart of the Steepest Descent method is depicted. With the Steepest Descent, we use a swap operator that swaps all of the orders with each other and then the best swap is taken as the new solution. This process repeats itself until there is no better solution possible. The reason we use a swap operator is because it is easy to implement and does not have a very high runtime. More information about the runtime is stated in *Section 5.3.* Whether a swap is beneficial for a schedule depends on what *Key Performance Indicator(or KPI)* is used to measure the value of the schedule. We will measure the schedule by using the KPIs average completion time and the maximum lateness of the orders in the schedule. In the Steepest Descent method, we combine these 2 KPIs into 1 objective function. These KPIs are combined because for ABI both KPIs are important. If we would only use 1 of the KPIs it could improve the schedule based on that KPI but have very bad results for the other KPI, and thus a bad result for the overall quality of the schedule. For instance, it is not desirable for ABI that a schedule is very efficient, but an order has a lateness of multiple weeks. Therefore, we calculate the KPI in the Steepest Descent method in the following way:

$$KPI = Average\ Completion\ Time + Maximum\ Lateness$$

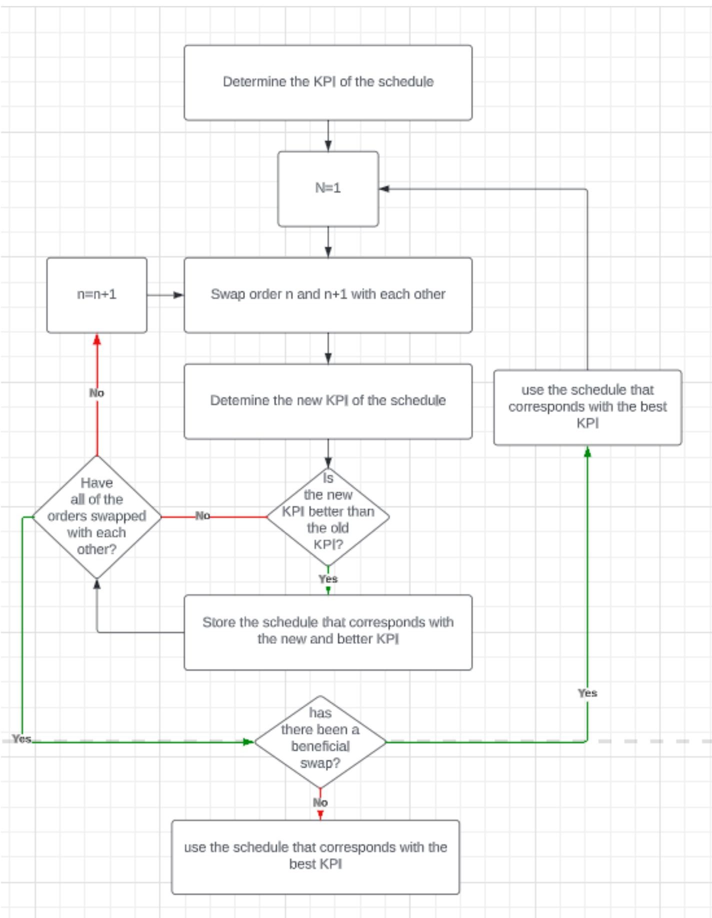

Figure 4-4: flowchart of the steepest descent method

When the KPI of the new schedule is lower it is considered better, and the swap is executed. We use the average value of the completion time because averages represent all of the orders in the data set. For instance, suppose there is 1 order with a very high release date, using the maximum completion time as a KPI, the schedule will not make any swaps since the total completion time does not improve. The reason we use the maximum lateness of the schedule is because when we use averages of the lateness a very early produces order can make up for a very late order.

4.4 Conclusion

In this section, we answer the research question:

- How can the scheduling situation and problem of ABI be classified, and how can this problem be solved?

The scheduling situation of ABI can be classified as a single-machine scheduling problem. The most important stage is the drilling and punching stage. ABI has 5 different drilling and punching machines, but based on the characteristics of the products in an order a single machine is chosen for production. To solve this problem we made a model to implement the priority rules Earliest Due Date and MinSlack. Since the schedule created by a priority rule often does not result in the optimal solution, we use the steepest descent method to improve the quality of the schedule based on the KPIs average completion time and maximum lateness.

5. Experiments

In this chapter, we discuss the experiments we do to find out what scheduling method works best for ABI. First, in *Section 5.1,* we discuss the experiments, the reason for these experiments, and how we set up the experiments. In *Sections 5.2* and *5.3,* we perform and analyze the experiments. Based on the results of the experiments we can make conclusions about what scheduling method is the best in the situation of ABI. In *Section 5.4,* we answer the research question for this chapter.

5.1 Setup of the Experiments

In *Chapter 4*, the method of solving the single-machine scheduling problem is described. These methods are the priority rules Earliest Due Date and MinSlack, both are implemented in MS Excel VBA to make a schedule. The code for the priority rules and the Steepest Descent method is located in *Appendix C*. For the experiments, we use 5 data sets that represent a normal situation for ABI with regard to their production schedule. The data sets are not all taken from reality but do represent it because ABI did not have that data available. These data sets are generated by manually looking up the orders that needed to be produced in a week. As explained in *Section 4.3.1,* the priority rules take into account the setup times between different types of heads. The 5 data sets can be found in *Appendix D* with more information about the head, release dates, and due dates of the orders. It is important to take multiple data sets, so, the results of the experiments can be compared with each other.

To test the influence of the setup time on the schedule we can use a weight to multiply the setup time and look at the influence of the different values for the weight of the setup time on the KPIs. Then, the priority value for the 2 priority rules earliest due date and minimum slack is calculated in the following way:

$$EDD = c_i - d_i - s_i * w$$
$$MinSlack = d_i - p_i - c_{i-1} + s_i * w$$

- the variables in the formula are stated in *Section 4.3.1*
- w = the weight given to the required setup time

The KPIs we use to determine the efficiency of the schedule are the average completion time, the maximum lateness, and the average lateness. In *Section 4.3.2*, the average completion time and maximum lateness are already explained why they are important to consider in the schedule. Even though, the average lateness can give a wrong representation of a good schedule by a very early order and a very late order to cancel each other out, because we also take into account the maximum lateness it does gives us important information. When the maximum lateness does not decrease, but the average lateness does, it means the schedule did improve.

Based on the results of the first experiment, the best value for w is used for the initial schedule. To improve the schedule, the Steepest Descent method is applied. The schedule is improved by looking at both the average completion time and the maximum lateness of a schedule. As mentioned in *Section 4.3.2,* these 2 KPIs are combined into 1 objective function to measure the efficiency of the schedule. In this experiment, we test the results of the KPIs after the Steepest Descent method is applied to the schedules made with the priority rules earliest due date, and minimum slack.

5.2 Experiments with the Weight of the Setup Time

"w" is the value of the weight which is multiplied by the setup time that is taken into account for the priority rule. The difference in weight for the setup time gives it a higher value in the priority rules. This means an order can get a higher priority and is scheduled through the setup time than it originally has with only the MinSlack or the EDD rule. A different value for the weight for the setup time can cause the average completion time, maximum lateness, or average lateness to change. For example, the results of the first data set using priority rule EDD for different values of "w" is given in *Table 2*.

Table 2: Data Set 1, MinSlack Rule

Data set 1 MinSlack Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	15.41	62.85	21.62
2	15.41	62.85	21.62
3	15.41	62.85	21.62
4	15.41	62.85	21.62
5	15.41	62.85	21.62
6	15.41	62.85	21.62
7	15.58	110.10	24.12
8	14.71	55.84	11.00
9	14.71	55.84	11.00
10	14.65	141.10	10.19

When the priority rule works with a higher weight for the setup times, the overall schedule becomes more efficient with regard to the total setup time, and thus the average completion time reduces. When the weight for the setup times is very high, all of the orders that are in the same *order family* will be clustered together. In the situation of ABI, an *order family* consists of all the orders with the same size head. When 2 different orders with the same order family are scheduled after each other there is no setup time. What is expected, is that the efficiency of the schedule increases to a certain value of the weight of the setup time after which it has a negative effect to cluster all of the orders in the same family together. This negative effect arises because orders that have a closer due date are scheduled later because they are not in the same order family. This will cause an increase in the lateness.

From the results of the experiment, we conclude the following. In *Table 3,* the results from the experiments are depicted, we show an overview of the values of the KPIs for the values w=1 and w=10. Both for the EDD and MinSlack priority rule, the general trend is decreasing for a higher value of "w" for the KPIs completion time and average lateness. The KPI maximum lateness is very volatile with different values for "w". The average completion time does not change much, in the data sets for both priority rules the decrease ranges from 0.1 to 4. The average lateness is more volatile, from the results we see a decrease ranging from 1.25 to 54.59. The KPI maximum lateness changes less consistently when the value for "w" increases. This is also influenced by the initial value the maximum lateness has for w=1. For example, in data set 2 for MinSlack the maximum lateness starts at 24.36 and increases to 74.87, an increase of 55.34 hours, while using the EDD rule maximum lateness started at 76.56 and decreased to 74.87.

Table 3: comparison of the KPIs for w=1 and w=10

Data set 1 MinSlack rule	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
w=1	15.41	62.85	21.62
w=10	14.65	141.10	10.19
Delta	0.76	-78,.25	11.43

Data set 1 EDD rule			
w=1	15.40	57.58	21.35
w=10	13.76	112.92	-3.24
Delta	1.64	-55.34	24.59
Data set 2 MinSlack rule			
w=1	16.27	24.36	-10.02
w=10	15.00	74.87	-26.59
Delta	1.27	-50.52	16.57
Data set 2 EDD rule			
w=1	19.17	76.56	27.70
w=10	15.00	74.87	-26.59
Delta	4.18	1.68	54.29
Data set 3 MinSack rule			
w=1	15.34	29.39	-14.10
w=10	15.05	29.39	-18.72
Delta	0.29	0.00	4.62
Data set 3 EDD-rule			
w=1	15.13	29.39	-17.47
w=10	15.05	29.39	-18.72
Delta	0.08	0.00	1.25
Data set 4 MinSlack rule			
w=1	16.65	46.64	11.41
w=10	16.28	78.38	5.84
Delta	0.37	-31.74	5.58
Data set 4 EDD rule			
w=1	16.52	43.35	9.41
w=10	15,.22	70,.43	-10.10
Delta	1.30	-27.08	19.51
Data set 5 MinSlack rule			
w=1	14.98	38.67	-31.87
w=10	13.57	8.08	-54.32
Delta	1.40	30.59	22.44
Data set 5 EDD rule			
w=1	14.12	0.88	-45.58
w=10	13.27	8.08	-59.19
Delta	0.85	-7.20	13.61

However, a comparison can be made between the 2 priority rules. For this comparison, we look at all of the results for the KPIs for the different values of "w". These tables can be found in *Appendix E*. To determine which priority rule delivers a better schedule we mostly look at the KPI average completion time and maximum lateness. The average completion time is the most important because this KPI determines how efficient a schedule is. The maximum lateness is also important because ABI does not want to have an order that is scheduled and delivered too late. The average lateness is less important but can be used to see how the schedule changes. Based on the KPIs MinSlack works better for Data Sets 1 and 2, and EDD works better for Data Sets 4 and 5. For Data Set 3 the KPIs, and therefore the schedule, are the same. In the data sets, the best value for "w" is often the same for both priority rules, except for Data Set 2. A reason for the decrease in the average completion time for an increasing value of "w" is that the due date is measured in whole days. This means the value for the weight of the setup time in the priority rule needs to be higher when there are multiple days between the due dates of orders to change the priority of the orders. Also, since we use a parallel generation scheme, some orders are not available for scheduling.

Since there is no clear value of "w" that results in the overall best schedule, we recommend that ABI tries different values for "w" that result in the best production schedule that satisfies their needs regarding the KPIs. We propose using the values w=4, w=8, and, w=10. When the outcomes of the KPIs are not as desired, other values can be tested. ABI can consider whether they want a more efficient schedule with a lower average completion time but with a higher maximum lateness, or a schedule that does not have the lowest average completion time with a moderate increase in the maximum lateness compared to the situation of w=1. The results of the experiments per priority rule and data sets can be seen in *Appendix E*.

5.3 Efficiency Priority Rules Compared to the Steepest Descent Methods

The KPIs are used to determine the efficiency of the schedule after applying the Steepest Descent method on the schedules made with the priority rules. The Steepest Descent method works with the KPIs average completion time and maximum lateness, as described in *Section 4.3.2*. The weight for the setup time is taken from the results of the experiment in *Section 5.2*. First, a schedule is made using only the priority rules, this schedule is the initial solution. Then, the Steepest Descent method is executed. *Table 4* shows the averages of the results of the experiment. In *Appendix F* the individual results per data set are given. The average computation time of the Steepest Descent method was almost 2 minutes, which is fine to make the schedule more efficient.

Table 4: Average results of the impact of the Steepest Descent

Averages of the experiment with the Steepest Descent			
Priority rule	Average Completion Time(Hour)	Maximum Lateness(Hour)	Average Lateness(Hour)
MinSlack	15.19	32.86	-12.98
Steepest Descent	14.59	21.55	-21.46
Delta	-0.60	-11.31	-8.48
%change	-3.97%	-34.41%	-65.34%
EDD	14.65	48.13	-20.62
Steepest Descent	14.26	34.39	-26.46
Delta	-0.40	-13.74	-5.84
%change	-2.71%	-28.55%	-28.31%

Based on the results from the experiment we can say the initial solution can be improved by using the Steepest Descent method. By using the Steepest Descent method, the schedule is improved from the initial solution that is made using only the priority rule based on the KPIs. This confirms the theory stated in *Section 3.3.1,* that priority rules are a good start but often do not result in the most efficient schedule. *Table 4* shows the EDD priority rule has a better performance with the KPIs average completion time and average lateness. However, for the KPI maximum lateness, the MinSlack priority rule provides better results. These results from the average outcomes of the KPIs are in line with the individual results of the performed experiment. Only for Data Set 3, the results are the same for all KPIs. This is not abnormal, since the initial solution, or schedule, is the same for both priority rules, so the same swaps are made. Also, important to note from this experiment is that the average completion time does not change a lot. This can be explained because the average completion time only changes when an order is swapped into its setup family. This means the order that is swapped does not have a setup time anymore and the average completion time goes down. The setup time is only a small part of the total production time, therefore, the decrease is relatively small. The KPI maximum lateness decreases by quite a lot, for both the MinSlack rule and the EDD rule. This big change in maximum lateness shows the importance of implementing the maximum lateness in the KPI we use in the Steepest Descent, and that the schedules can very much improve regarding the maximum lateness. The average lateness is the least important KPI, it is also not taken into account when using the Steepest Descent improvement method. This is because we look at an average, meaning a very early order can compensate for a very late order. However, the KPI is interesting to look at because it shows while the average completion time, and the maximum lateness decrease, the lateness per order also decreases. The relation between a decreasing maximum lateness and a decreasing average lateness is normal. But, the average lateness decreases more than the effect of the decrease in maximum lateness, indicating the lateness of the schedule is reduced.

5.4 Conclusion

In this section, we provide an answer to the following research question based on the findings of *Chapter 5.*

- What way of scheduling works the best for ABI, and how can the schedule be further improved?

To determine what priority rule works best for ABI we need to test how the earliest due date and minimum slack perform with data sets that represent the scheduling situation of ABI. Because the orders have a setup time if the previous order was not in the same order family also an experiment needs to be done to see what the effect is when the weight of the setup time in the priority rules is increased. To see what priority rule works better 3 KPIs are chosen, the average completion time, maximum lateness, and, average lateness. Based on these KPIs we can determine which scheduling method is better and results in a more efficient schedule. The priority values of the orders are calculated like this:

$$EDD = c_i - d_i - s_i * w$$
$$MinSlack = d_i - p_i - c_{i-1} + s_i * w$$

Based on the results from this experiment we cannot say which priority rule is better. For half of the data sets the EDD rule works better, and for the other half the MinSlack rule. For the value of the weight for the setup time, the results show that we cannot exactly say what value results in the best schedule. However, different values for the weight can be tried, and based on the value of the KPIs a value can be chosen.

The second experiment tested the combination of the priority rules with the improvement method Steepest Descent. To answer the second part of the research question, yes the initial schedule made with the priority rules can be improved, based on the change in KPIs after using the Steepest Descent. For the result, we take the average of the results from all of the experiments we performed. The EDD rule performs better with the KPIs average completion time and average lateness, and the MinSlack rule with the KPI maximum lateness.

6. Conclusion

In this chapter, the findings of the research are discussed. *Section 6.1* presents the main findings to form a recommendation for ABI. *Section 6.2*, provides the discussion and limitations of the research. Lastly, we discuss future research that can be conducted in *Section 6.3*.

6.1 Main Findings and Recommendations

In *Chapter 1,* we describe the problem and formulate this into a central research question, supported by sub-research questions. We go over the findings in this thesis to answer the central research question of this thesis:

How can ABI get more insight into the production scheduling process to improve its production schedule?
The first steps are taken in *Chapter 2* and *Chapter 3*, by gathering information about the current situation of ABI and consulting the literature about machine scheduling and scheduling methods. In *Chapter 2*, the current situation is mapped with the production process and its characteristics, like what machines are used and the processing speeds of the machines. From *Chapter 2*, it becomes clear that ABI needs a production schedule, since there is no clear method to schedule their order, and the available data shows they have a lot of backorders. In *Chapter 3*, different kinds of machine scheduling environments, scheduling methods, and improvement methods are described to find information about a scheduling method suitable for ABI.

Important findings for ABI are from *Chapter 4*, here the information gathered in *Chapter 2* and *Chapter 3* are combined, to make decisions about what stage in the production process is modeled and in what method is used to make the production schedule. The drilling and punching stage of ABI's operation is the most important stage. This is because this stage is the bottleneck of the production process and the whole operation can revolve around the drilling and punching stage, meaning the other stages in the production process can follow the same production schedule. Also, the drilling and punching stage is important because every product needs to go through this stage, which is not always the case for the other stages. This stage is the only stage with setup times, meaning the sequence in which the orders are scheduled on this machine influences the makespan of the production process. The drilling and punching stage is described as a single-machine scheduling problem. This is important to know since it influences what scheduling methods can be used and how the situation is modeled.

There are numerous ways to make a production schedule. In the current situation, ABI does not have a consistent way of scheduling. So, we choose a fairly straightforward scheduling method. We implement the priority rules earliest due date and minimum slack. Because priority rules are simple to use and implement, the initial schedule is often not the most efficient. The improvement method we use is the Steepest Descent method. This method uses a KPI which minimizes the average completion time and minimizes the maximum lateness, by combining the 2 into 1 KPI to measure the efficiency of a schedule.

In *Chapter 5*, we tested the priority rules by looking at the KPIs average completion time, maximum lateness, and, average lateness. For the first experiment, we used different weights for the setup time and did experiments to test the impact of the Steepest Descent on the measured KPIs. The conclusion for the weight for the setup time could not be exactly given. So, for the weight of the setup time we recommend first to test the values w=4, w=8, and, w=10. When the desired values for the KPIs are not found at these values for "w" other values can be tested, until ABI finds the schedule with the desired values for the KPI. In this experiment, it is a tie between the 2 priority rules. This means it depends per data set which priority rule performs better.

The result following the second experiment is that the Steepest Descent method improves on the initial schedule made with the priority rules. The average relative change for implementing the Steepest Descent method on the MinSlack rule is for the KPIs average completion time -3,97%, maximum lateness -34,41%, and, average lateness -65,34%. For the EDD rule this is -2,71%, -28,55%, and, -28,31% respectively. Based on the values for the KPIs the EDD rule performs better with the KPIs average completion time and average lateness, while the MinSlack rule the value for the maximum lateness is better.

6.2 Discussion and Limitations

In this research, two aspects are not as we wanted them to be that could influence the research we have done. The two aspects are the data sets and the theoretical model. The data sets we used with information about the orders that need to be scheduled are not all from real-life data, since that data is not stored. Also, the sequence and processing times of each order are not stored. This means the effect on the efficiency of the schedules after applying the priority rules or the Steepest Descent cannot be compared with each other. Because this data of the orders is not stored, the processing time of the machines is an estimation. This estimation of the processing time is used in the theoretical model, and in theory that time is correct. However, in this estimation factors like irregular (bathroom) breaks are not included. Another limitation is the number of data sets used, preferably more data sets were used, which came from real data. That data would be more reliable and the results of the experiments we did on those data sets could say more than the results of the experiments we have done now. However, this was inevitable, and the data we did have available and used could be compared to each other and were a good enough representative of real data.

The second aspect is the theoretical model we made and used for implementing the priority rules and Steepest Descent method. The model is straightforward: maintenance, breakdowns, sickness of the employee, and, other things that influence the processing time of the machine are not taken into account. The model is theoretical, when it is used in practice and variables like the release date of an order is changed, it can have a big effect on the entire schedule. Also, the model only looks at the drilling and punching stage. So, when the other stages do not work faster than the drilling and punching stage, thus creating idle time, or are under maintenance, it cannot be taken into account in the model.

In this research, it is chosen to look at the KPIs average completion time, maximum lateness, and, average lateness. These KPIs are the most important for ABI, but the research could have extended into more KPIs with the effect of the priority rules and Steepest Descent method. We looked into priority rules and the Steepest Descent method. These scheduling methods are relatively straightforward, easy to implement, and good to use for a medium-sized company like ABI, that in the first place did not have a production scheduling method. However, more advanced scheduling methods like mathematical models where uncertainty can be implemented or other heuristics like simulated annealing can be interesting to implement in the scheduling process to improve on the KPIs of the schedule.

6.3 Future Research

Right now, the production schedule is made with orders already in the system. ABI can look into forecasting incoming orders. However, the orders ABI produces can be very different from each other, with regard to the specifications of the brushes. So, forecasting could be done for regular customers that order the same products in each order. With forecasting, ABI can anticipate orders and realize a shorter delivery time. When forecasting orders that require lats, the release date of that order can be brought back. This means processing the order can start earlier, decreasing the delivery time of the order.

To further improve the production schedule, the model we made for this research can be expanded upon. The model can be adjusted to resemble reality better, by implementing the possibility for breakdowns, maintenance, and, other occurrences that slow down the production process. The model can be expanded further by researching factors that influence the production process. It could be the case that machines have a slower processing speed for certain types of products. Providing information about the different processing speeds per type of product gives a better insight into the delivery time, which ABI can use to communicate a more accurate delivery time to their customer. The type of scheduling method used can be explored. With this information, different objectives for the schedule can be tested. Then, the best objective can be chosen for each schedule making the operation more diversifiable. Besides different scheduling methods, more complex metaheuristics can be implemented to increase the efficiency of the schedule, like the taboo search algorithm or simulated annealing.

The whole production process of ABI could be mapped. This way, the influences of the different stages on each other can be discovered and the whole operation can be adjusted to each other. Then, the entire production process can be looked at as a whole, and different ways of scheduling can be implemented to fit with the new machine scheduling situation. The different scheduling methods can be compared and the one that is the most beneficial for the entire production process can be used. Another way of making the production process more efficient is to implement machine allocation in the model. Right now, the orders are allocated to a machine by the employees through experience. When the machine allocation is combined with production scheduling the machines could be operating more efficiently. Besides, when employees leave or retire the knowledge leaves with them.

Bibliography

Attea, B., Abbood, A., Hasan, A., Pizzutti, C., Al-Ani, M., Ozdemir, S., & Al-dabbagh, R. (2021). *A review of heuristics and metaheuristics for community detection in complex networks: current usage, emerging development and future directions.*

Balas, E., Simonetti, N., & Vazacopoulos, A. (2008). *Job shop scheduling with setup times, deadlines and precedence constraints.*

Cheng, T., & Sin, C. (1990). A state-of-the-art review of parallel machine scheduling research. *European Journal of Operational Research* , 271-292.

Cooper, D. R., & Schindler, P. S. (2014). *Business research methods.* McGraw-Hill.

de Boer, R. (1998). *Resource Constrained Multi Project Management A hierarchicaldecision support system.* Enschede: febodruk.

FAN, L. (2020). manufacturing job shop scheduling problems based on improved meta-heuristic algorithm and bottleneck identification. *academic journal of manufacturing engineering*, 98-103.

Fayaed, S., El-Shafie, A., & Jaafar, O. (2013). *reservoir-system simulation and optimization techniques.*

Glover, F., & Sorensen, K. (2015). *Metaheuristics.*

Heerkens, H., & van Winden, A. (2017). *Solving Managerial Problems Systematically.* Noordhoff.

Lee, K.-M., & Yamakawa, T. (1998). *A genetic algorithm for general machine scheduling problems.*

Liaee, M. M., & Emmons, H. (1997). scheduling families of jobs with setup times. *international journal of production economics*, 165-176.

Naderi, B., Zandieh, M., Khaleghi Ghoshe Balagh, A., & Roshanaei, V. (2009). An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness. *Expert systems with Applications*, 9625-9633.

Sanlaville, E., & Schmidt, G. (1998). machine scheduling with availability constraints. *Acta informatica*, 795-811.

Schutten, M. (1996). *shop floor scheduling with setup times.* Enschede: Febodruk.

Senthilkumar, P., & Narayanan, S. (2010). Literature review of single machine scheduling problem with uniform parallel machines. *Intelligent information management*, 457-474.

Vanhoucke, M. (n.d.). *Integrated project management sourcebook.* Springer.

Appendix A – Data Weekly Number of Orders and Products

Table 5: Data Weekly Number of Orders and Products

week	# of products too late	# of total products	% too late	# of orders too late	# of total orders	% too late
12	1137	1525	74,56%	10	42	23,81%
13	488	759	64,30%	28	53	52,83%
14	284	564	50,35%	17	31	54,84%
15	272	710	38,31%	10	33	30,30%
16						
17	752	1219	61,69%	34	54	62,96%
18						
19						
20	1085	1575	68,89%	55	81	67,90%
21	1494	2405	62,12%	70	88	79,55%
22						

23	1423	1564	90,98%	86	112	76,79%
24						
25						
26						
27	1009	2654	38,02%	63	82	76,83%
28	2245	2932	76,57%	36	65	55,38%
29	2363	2566	92,09%	38	54	70,37%
30	1338	1544	86,66%	23	33	69,70%
31						
32						
33						
34						
35	580	877	66,13%	16	36	44,44%
36						
37	233	496	46,98%	13	27	48,15%
38	379	640	59,22%	24	32	75,00%
39	117	477	24,53%	18	24	75,00%
40	440	502	87,65%	18	27	66,67%
41	353	977	36,13%	10	20	50,00%
42	584	1466	39,84%	7	25	28,00%
43	813	1258	64,63%	13	35	37,14%
44	355	419	84,73%	15	34	44,12%
45	487	870	55,98%	24	47	51,06%
46	812	987	82,27%	43	54	79,63%
47	350	407	86,00%	18	28	64,29%
48	160	461	34,71%	16	34	47,06%
49	315	462	68,18%	19	35	54,29%
50	152	167	91,02%	22	30	73,33%
average	960,70	1378,35	68,01%	39,05	59,58	60,45%

The blue rows show a week where data is not collected and the yellow rows depict the holidays in which there is no production.

Appendix B – Data weekly Number of Orders and Products per Machine

Table 6: Data Weekly Number of Orders and Products per Machine

	Machine1		Machine2		Machine3		Machine4		Machine5	
week	Orders	Products	Orders	Products	Orders	Products	Orders	Products	Orders	Products
12	11	712	20	435	1	20	-	-	4	148
13	16	404	17	140	6	76	-	-	2	40
14	7	119	8	121	3	34	-	-	1	112
15										
16										
17	14	490	11	280	-	-	-	-	10	222
18										
19										

20	19	648	13	450	11	203	-	-	9	30
21	17	381	18	1159	12	225	-	-	10	40
22										
23	19	286	28	555	11	193	2	100	7	30
24										
25										
26										
27	16	594	32	744	6	53	2	1050	1	30
28	10	441	29	1006	1	20	1	1000	10	240
29	9	474	24	647	-	-	1	1000	8	204
30	5	159	11	93	-	-	1	1000	6	172
31										
32										
33										
34										
35	3	69	10	197	7	70	1	288	6	155
36										
37	7	141	12	331	6	76	1	200	2	72
38	4	7	10	261	3	9	1	200	1	16
39	4	306	8	60	4	15	-	-	-	-
40	6	1100	16	194	1	6	-	-	1	5
41	2	794	12	755	5	86	1	50	-	-
42	4	569	13	769	4	80	1	50	8	184
43	3	75	14	861	9	139	1	50	8	224
44	3	75	7	233	5	67	1	50	1	2
45	5	42	13	356	5	67	-	-	2	32
46	7	132	17	395	6	68	-	-	3	44
47	3	88	9	172	4	25	-	-	1	4
48	4	142	17	317	2	5	-	-	6	92
49	5	144	9	207	3	15	-	-	1	12
50	3	20	3	37	1	3	-	-	-	-
Average	11,374	349,687	16,42	434,918	6,746	100,675	1,25	330,1	5,167	83,9688
MIN	2	7	3	37	1	3	1	50	1	2
MAX	19	1100	32	1159	12	225	2	1050	10	240

The blue rows show a week where data is not collected and the yellow rows depict the holidays in which there is no production.

Appendix C – Pseudocode of the Priority Rules and Steepest Descent Method

MinSlack Priority rule

```
'onderstaande sub alleen uitvoeren in de sheet OrderSheet
Sub MinSlack()
    Dim lastrow As Integer
    Dim k As Integer
    Dim z As Integer
    Dim x As Integer
    Dim j As Integer
    Dim m As Integer
    Dim n As Integer
    Dim i As Integer
    Dim y As Integer
    Dim setupTime As Double
    Dim Slack As Double
    Dim completionTime As Double
    Dim Rng1 As Range
    Dim Rng2 As Range
    Dim NewRng As Range
    Dim MinVal As Double
    Dim MinCell As Range
    Dim MinRow As Integer
    Dim tasks As Variant
    Dim lateness As Double

m = 1 'Multiplication factor of the setup time
lastrow = Range("A" & Rows.Count).End(xlUp).Row

'delete the rows of completionTime and lateness
    For i = 2 To lastrow
        Cells(i, 11).Value = "" 'deletes the column completion times
        Cells(i, 12).Value = "" 'deletes the column lateness
    Next i

completionTime = WorksheetFunction.Min(Cells(2, 8), Cells(lastrow, 8)) 'set completionTime as the lowest release date

For k = 1 To lastrow 'Loop over every order

'if the completionTime is calculated it means the order is scheduled and no order can be scheduled on that row
'checks whether an order is scheduled to assign value to the variable z to calculate the lateness based on the last completionTime
z = 2
    For x = 2 To lastrow
        If IsEmpty(Cells(x, 11)) Then
            Cells(x, 13).Value = 0
                Else
            Cells(x, 13).Value = 1
        End If

        If Cells(x, 13).Value = 1 Then
            z = z + 1
        Else
            z = z
        End If
    Next x

'make the range dynamic, so the right amount of orders is taken into account for scheduling
    Set Rng1 = Cells(z, 12) 'use z so the lateness of the already scheduled orders is not calculated
    Set Rng2 = Cells(lastrow, 12)
    Set NewRng = Range(Rng1, Rng2)

'Calculate the prio(slack) while making sure the orders with a releasedate>completion time aren't scheduled
    For j = 2 To lastrow
        If Cells(j, 8).Value > completionTime Then 'completionTime moet een waarde hebben
            Slack = 1000 'so this order has a very high slack and won't be scheduled
        ElseIf z = 2 Then 'z needs to start at 2 because of the header
            Slack = Cells(j, 10).Value - Cells(j, 9).Value - completionTime
        ElseIf Cells(j, 13).Value = 1 Then  'if the order is scheduled the lateness is calculated
            Slack = Cells(j, 11).Value - Cells(j, 10).Value 'the lateness is calculated, needed for data
        Else    'if the order is not yet scheduled, the slack is determined using the completionTime of the last order that is scheduled
            If Cells(z - 1, 6).Value <> Cells(j, 6) Then
                setupTime = ((3 / 24) * 4.2) * m 'value setup time
            Else
```

```
            ----
                setupTime = 0
            End If
            Slack = Cells(j, 10).Value + setupTime - Cells(z - 1, 11).Value - Cells(j, 9).Value
        End If
        Cells(j, 12) = Slack 'assign the calculated value of the slack to the cell
    Next j

'In this part the scheduling takes place
'find the min value in the column of "Slack"
    MinVal = WorksheetFunction.Min(NewRng) 'find the minimum value of the range
    'use the Find function to get information about MinCell and us the Row number
    Set MinCell = NewRng.Find(what:=MinVal)

    If MinVal = 1000 Then 'if all of the releaseDates are bigger than the completionTime then the order with the lowest releaseDate is put first
        tasks = Range(Cells(z, 1), Cells(21, 13)).Value
        tasks = Application.WorksheetFunction.Sort(tasks, 8, True) 'sort the releaseDates from low to high
        Range(Cells(z, 1), Cells(21, 13)).Value = tasks
    ElseIf MinCell.Row = z Then 'the order with the highest prio is already in the good row
        completionTime = completionTime + Cells(z, 9).Value
    Else 'cut and insert the row with the highest prio to the first available row
        Range(Cells(MinCell.Row, 1), Cells(MinCell.Row, 12)).Select
        Selection.Cut
        Cells(z, 1).Select
        Selection.Insert Shift:=xlDown 'shift all the remaining orders down
        completionTime = completionTime + Cells(z, 9).Value
    End If

'give value to the cells in the column completionTime
    If Cells(z, 8).Value > completionTime Then 'if the releaseDate is bigger than the completionTime the next completionTime is ReleaseDate+processingTime
        completionTime = Cells(z, 8).Value + Cells(z, 9)
    End If

        Cells(z, 11).Value = completionTime
Next k

'here the setup time has to be added permanently to the completionTime
    If Cells(2, 8).Value > Cells(2, 11).Value Then 'if the releaseDate is bigger than the completionTime the next completionTime is ReleaseDate+processingTime
        completionTime = Cells(2, 8).Value + Cells(2, 9).Value + (1 / 24 * 4.2) 'include a setupTime for the first order scheduled
    Else
        completionTime = Cells(2, 11).Value + (1 / 24 * 4.2) 'include a setupTime for the first order scheduled
    End If
        lateness = completionTime - Cells(2, 10).Value 'New lateness because of new compeltionTime
        Cells(2, 11).Value = completionTime
        Cells(2, 12).Value = lateness

    For y = 3 To lastrow
        If Cells(y - 1, 6).Value <> Cells(y, 6).Value Then
            setupTime = (1 / 24) * 4.2
        Else
            setupTime = 0
        End If

        If Cells(y, 8).Value > completionTime Then 'if the releaseDate is bigger than the completionTime the next completionTime is ReleaseDate+processingTime
            completionTime = Cells(y, 8).Value + Cells(y, 9).Value + setupTime
        Else
            completionTime = completionTime + Cells(y, 9).Value + setupTime 'calculate new completionTime with the setup time

        End If

        lateness = completionTime - Cells(y, 10).Value 'because of the new completionTime a new lateness is calculated
        'assign the values to the cells
        Cells(y, 11).Value = completionTime
        Cells(y, 12).Value = lateness
    Next y

    Cells(z, 11).Value = "" 'deletes contents of this cell because z goes 1 too far because it starts at 2 and adds all orders

    For i = 2 To lastrow

        Cells(i, 13).Value = "" 'deletes the column which keeps up if an order is scheduled
    Next i

End Sub
```

Priority rule EDD

```
Sub EDD()
    Dim lastrow As Integer
    Dim k As Integer
    Dim z As Integer
    Dim x As Integer
    Dim j As Integer
    Dim m As Double
    Dim n As Integer
    Dim i As Integer
    Dim y As Integer
    Dim lateness As Double
    Dim setupTime As Double
    Dim completionTime As Double
    Dim Rng1 As Range
    Dim Rng2 As Range
    Dim Rng3 As Range
    Dim Rng4 As Range
    Dim NewRng2 As Range
    Dim NewRng As Range
    Dim MaxVal As Double
    Dim MaxRow As Integer
    Dim MaxCell As Range
    Dim tasks As Variant

m = 8 'multplication factor of the setup time
lastrow = Range("A" & Rows.Count).End(xlUp).Row

'delete the rows of completionTime and lateness
    For i = 2 To lastrow
        Cells(i, 11).Value = "" 'deletes the column completion times
        Cells(i, 12).Value = "" 'deletes the column lateness
    Next i

completionTime = WorksheetFunction.Min(Cells(2, 8), Cells(lastrow, 8)) 'set completionTime as the lowest release date
For k = 1 To lastrow 'Loop over every order


'if the completionTime is calculated it means the order is scheduled and no order can be scheduled on that row
'checks whether an order is scheduled to assign value to the variable z to calculate the lateness based on the last completionTime
z = 2
    For x = 2 To lastrow
        If IsEmpty(Cells(x, 11)) Then
            Cells(x, 13).Value = 0
                Else
            Cells(x, 13).Value = 1
        End If
        If Cells(x, 13).Value = 1 Then
            z = z + 1
        Else
            z = z
        End If
    Next x

    Set Rng1 = Cells(z, 12)  'use z so the lateness of the already scheduled orders is not calculated
    Set Rng2 = Cells(lastrow, 12)
    Set NewRng = Range(Rng1, Rng2) 'use more variables so the range is dynamic

    Set Rng3 = Cells(2, 10)
    Set Rng4 = Cells(lastrow, 10)
    Set NewRng2 = Range(Rng3, Rng4)

'Calculate the lateness while making sure the orders with a releasedate(column 8) aren't scheduled
    For j = 2 To lastrow
        If Cells(j, 8).Value > completionTime Then
            lateness = -1000  'so this order has a very low lateness and won't be scheduled in the next step
        ElseIf z = 2 Then 'z needs to start at 2 because of the header
        lateness = completionTime - Cells(j, 10).Value 'no setupTime because it is the first order to be scheduled

        ElseIf Cells(j, 13).Value = 1 Then  'if the order is scheduled the lateness is equal to completionTime-dueDate

        lateness = Cells(j, 11).Value - Cells(j, 10).Value

        Else 'calculate lateness for orders that are not yet scheduled
            If Cells(z - 1, 6).Value <> Cells(j, 6) Then 'take into account the setup time
```

```
                setupTime = ((3 / 24 * 4.2) * m) 'priority value of setup time
                Else
                setupTime = 0
            End If

'if the order is not yet scheduled,the lateness is determined using the completionTime of the last order that is scheduled
        lateness = Cells(z - 1, 11).Value - setupTime - Cells(j, 10).Value '-setupTime because then it gets less prio to be scheduled
        End If
        Cells(j, 12) = lateness
    Next j

'find the max value in the column of "lateness"
    MaxVal = WorksheetFunction.Max(NewRng) 'find the maximum value of the range

    'use the Find function to get information about MaxCell and us the Row number
    Set MaxCell = NewRng.Find(what:=MaxVal) ' , LookIn:=xlValues, MatchCase:=False)

'de order met de hoogste prio wordt niet ingepland, maar de orders worden van release date van laag naar hoog gepland
    If MaxVal = -1000 Then   'if all of the releaseDates are bigger than the completionTime then the order with the lowest releaseDate is put first
'als de releaseDates hetzelfde zijn moet er wel naar due date gekeken worden
        tasks = Range(Cells(z, 1), Cells(lastrow, 13)).Value
        tasks = Application.WorksheetFunction.Sort(tasks, 8, True) 'sort the releaseDates from low to high
        Range(Cells(z, 1), Cells(lastrow, 13)).Value = tasks
    ElseIf MaxCell.Row = z Then 'the order with the highest prio is already in the good row
        completionTime = completionTime + Cells(z, 9).Value
    Else 'cut and insert the row with the highest prio to the first available row
        Range(Cells(MaxCell.Row, 1), Cells(MaxCell.Row, 12)).Select
        Selection.Cut
        Cells(z, 1).Select
        Selection.Insert Shift:=xlDown 'shift all the remaining orders down
        completionTime = completionTime + Cells(z, 9).Value
    End If

'give value to the column completionTime
    If Cells(z, 8).Value > completionTime Then 'if the releaseDate is bigger than the completionTime the next completionTime is ReleaseDate+processingTime
        completionTime = Cells(z, 8).Value + Cells(z, 9)

    End If

    Cells(z, 11).Value = completionTime
Next k

'here the setup time has to be added permanently to the completionTime
'The value regarding the setupTime in the other setupTime related code can be changed for prio but not here
    If Cells(2, 8).Value > Cells(2, 11).Value Then 'if the releaseDate is bigger than the completionTime the next completionTime is ReleaseDate+processingTime
        completionTime = Cells(2, 8).Value + Cells(2, 9).Value + (1 / 24 * 4.2) 'include a setupTime for the first order scheduled
    Else
        completionTime = Cells(2, 11).Value + (1 / 24 * 4.2) 'include a setupTime for the first order scheduled
    End If
        lateness = completionTime - Cells(2, 10).Value 'New lateness because of new compeltionTime
        Cells(2, 11).Value = completionTime
        Cells(2, 12).Value = lateness

    For y = 3 To lastrow
        If Cells(y - 1, 6).Value <> Cells(y, 6).Value Then
            setupTime = (1 / 24 * 4.2)
        Else
            setupTime = 0
        End If

        If Cells(y, 8).Value > completionTime Then 'if the releaseDate is bigger than the completionTime the next completionTime is ReleaseDate+processingTime
            completionTime = Cells(y, 8).Value + Cells(y, 9).Value + setupTime
        Else
            completionTime = completionTime + Cells(y, 9).Value + setupTime 'calculate new completionTime with the setup time

        End If

        lateness = completionTime - Cells(y, 10).Value 'because of the new completionTime a new lateness is calculated
        'assign the values to the cells
        Cells(y, 11).Value = completionTime
        Cells(y, 12).Value = lateness

    Next y

    Cells(z, 11).Value = "" 'deletes contents of this cell because z goes 1 too far because it starts at 2 and adds all orders

    For i = 2 To lastrow
        Cells(i, 13).Value = "" 'deletes the column which keeps up if an order is scheduled
    Next i

End Sub
```

Steepest Descent method

```
Sub SteepestDescent()

Dim i As Integer
Dim j As Integer
Dim y As Integer
Dim x As Integer
Dim tasksOld As Variant
Dim tasksNew As Variant
Dim Rng1 As Range
Dim Rng2 As Range
Dim Rng3 As Range
Dim Rng4 As Range
Dim Rng5 As Range
Dim Rng6 As Range
Dim LRng As Variant
Dim NewRng As Range
Dim Arr1 As Variant
Dim Arr2 As Variant
Dim KPIOld As Double
Dim KPINew As Double
Dim KPIBest As Double
Dim BestRng As Variant
Dim LastRow As Integer
Dim completionTime As Double
Dim lateness As Double
Dim setupTime As Double

LastRow = Range("A" & Rows.Count).End(xlUp).Row
Set Rng3 = Cells(2, 11)
Set Rng4 = Cells(LastRow, 11)
Set Rng5 = Cells(2, 12)
Set Rng6 = Cells(LastRow, 12)
Set NewRng = Range(Rng3, Rng4)
Set LRng = Range(Rng5, Rng6)
'set the KPI as the value it has currently with the initial prio rule
KPIOld = (WorksheetFunction.Average(NewRng) / (LastRow - 1)) + WorksheetFunction.Max(LRng) 'average completion time

'loop until there is no better swap
Do

For j = 2 To LastRow
    For i = j + 1 To LastRow
'Swap 2 rows with each other
    Set Rng1 = Range(Cells(i, 1), Cells(i, 12))
    Set Rng2 = Range(Cells(j, 1), Cells(j, 12))

    Arr1 = Rng1.Value 'assign the array to a row
    Arr2 = Rng2.Value 'assign the second array to a row
'make the swap
    Rng1.Value = Arr2 'assign the values of the first array to the second row
    Rng2.Value = Arr1 'assign the values of the second array to the first row

'Calculate the new values for the completiontime, with the setup times and the lateness
'if a swap happened the completionTime is still from before the swap, so higher than the releaseDate
        completionTime = Cells(2, 8).Value + Cells(2, 9).Value + Worksheets("Gegevens").Range("A21").Value
        lateness = completionTime - Cells(2, 10).Value 'New lateness because of new compeltionTime
        Cells(2, 11).Value = completionTime
        Cells(2, 12).Value = lateness

'this does the same for row 2 but then in a loop for all of the rows
    For y = 3 To LastRow
        If Cells(y - 1, 6).Value <> Cells(y, 6).Value Then
            setupTime = (1 / 24 * 4.2) 'half an hour setup time divided by 24 hours in a day
        Else
            setupTime = 0
        End If

        If Cells(y, 8).Value > completionTime Then 'if the releaseDate is bigger than the completionTime the next completionTime is ReleaseDate+processingTime
            completionTime = Cells(y, 8).Value + Cells(y, 9).Value + setupTime
        Else
            completionTime = completionTime + Cells(y, 9).Value + setupTime 'calculate new completionTime with the setup time
        End If

        lateness = completionTime - Cells(y, 10).Value 'because of the new completionTime a new lateness is calculated
        Cells(y, 11).Value = completionTime 'assign the values to the cells
        Cells(y, 12).Value = lateness
    Next y
```

```
'calculate the new value for the KPI, if it is better swap keep the schedule if not swap the rows back and store the value of the cells
KPINew = (WorksheetFunction.Average(NewRng) / (LastRow - 1)) + (WorksheetFunction.Max(LRng))
tasksNew = Range(Cells(2, 1), Cells(LastRow, 12)).Value

    If KPINew < KPIOld Then
        BestRng = Range(Cells(2, 1), Cells(LastRow, 12)).Value 'the range with the swap is stored
        Range(Cells(2, 1), Cells(LastRow, 12)).Value = tasksOld 'the swap is reversed

        KPINew = KPIOld    'The new KPI needs to be compared with the next KPI of the next swap
    Else
        Range(Cells(2, 1), Cells(LastRow, 12)).Value = tasksOld 'if the KPI is not better the swap needs to be reversed to the old schedule
    End If
    Next i 'go to next swap
Next j  'go to next swap

'schedule is changed(or not) according to the best swap
    If IsEmpty(BestRng) Then
        Range(Cells(2, 1), Cells(LastRow, 12)).Value = tasksOld
     Else
        Range(Cells(2, 1), Cells(LastRow, 12)).Value = BestRng
        BestRng = tasksOld
    End If

 Loop Until IsEmpty(BestRng)

End Sub
```

Appendix D – Data Sets

Table 7: Data Set 1

OrderNumber	Number of Products	Head	Release Date	Processing Time	Due Date
Order1	40	10	0	1,152	-1
Order13	140	10	0	1,728	4
Order2	3	7	0	0,046	2
Order12	125	5	0	2,033	4
Order15	75	7	3	1,260	5
Order3	1	7	0	0,140	7
Order9	10	7	1	0,108	7
Order8	135	7	1	2,888	10
Order7	12	5	2	0,390	8
Order11	115	10	0	1,497	11
Order4	2	7	4	0,145	12
Order14	150	5	0	2,264	15
Order6	20	5	2	1,033	14
Order5	2	7	4	0,075	16
Order10	60	7	7	0,858	17

Table 8: Data Set 2

OrderNumber	Number of Products	Head	Release Date	Processing Time	Due Date
Order1	10	7	0	0,4533	1
Order2	60	7	0	1,8375	4
Order5	135	7	0	1,4950	7
Order3	20	5	0	0,8085	3
Order4	12	5	0	0,4925	6
Order6	30	10	2	0,6983	8
Order8	30	10	5	0,6983	11
Order7	60	10	6	1,3965	12
Order10	20	10	0	0,4043	13

Order13	4	7	0	3,2487	12
Order12	9	7	0	2,0507	15
Order9	20	5	6	0,6615	12
Order11	30	10	10	0,6983	16

Table 9: Data Set 3

OrderNumber	Number of Products	Head	Release Date	Processing Time	Due Date
Order1	100	3	0	2,2750	3
Order4	30	5	0	1,3160	3
Order2	10	5	0	0,9072	4
Order8	30	5	0	1,3766	5
Order3	10	5	2	0,8960	7
Order16	75	5	4	0,5484	8
Order15	75	5	0	0,5484	9
Order5	30	5	3	1,3230	11
Order7	1	5	5	0,0840	13
Order13	4	5	0	0,3927	18
Order6	30	3	3	1,2600	11
Order9	3	3	6	0,3360	14
Order10	10	3	7	0,9030	15
Order11	20	3	8	1,8025	16
Order14	40	3	0	2,2946	19
Order12	10	5	10	1,0507	17

Table 10:Data Set 4

OrderNumber	Number of Products	Head	Release Date	Processing Time	Due Date
Order2	20	7	0	0,248888889	1
Order1	75	7	0	2,8	4
Order3	35	5	0	0,544444444	2
Order4	40	5	0	1,132444444	4
Order5	70	5	2	0,435555556	6
Order10	120	5	0	2,24	9
Order15	75	7	3	0,7	10
Order14	80	7	0	1,493333333	12
Order9	60	7	7	0,56	14
Order8	30	7	7	0,093333333	14
Order13	200	10	0	2,986666667	11
Order6	100	10	5	0,777777778	13
Order7	150	10	5	0,7	15
Order11	50	5	0	0,855555556	16
Order12	10	5	0	0,080888889	18

Table 11: Data Set 5

OrderNumber	Number of Products	Head	Release Date	Processing Time	Due Date
Order4	400	7	0	1,208333333	2
Order1	150	7	0	0,90625	3
Order2	30	7	2	0,3171875	9
Order6	120	10	0	1,16	5
Order7	80	10	0	1,305	8
Order8	20	10	0	0,3625	9
Order15	75	7	3	1,8125	8
Order5	25	7	5	0,377604167	12
Order9	200	5	3	1,0875	10
Order16	10	5	0	0,208333333	12
Order10	15	5	8	0,06796875	15
Order11	175	5	0	1,26875	16
Order13	80	5	0	1,691666667	16
Order3	300	7	10	3,2625	17
Order14	90	7	0	2,28375	20
Order12	50	5	12	0,966666667	19

Appendix E – KPI Results for Different Weights of the Setup Time

Table 12: Data Set 1 MinSlack Rule

Data set 1 MinSlack Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	15.41	62.85	21.62
2	15.41	62.85	21.62
3	15.41	62.85	21.62
4	15.41	62.85	21.62
5	15.41	62.85	21.62
6	15.41	62.85	21.62
7	15.58	110.10	24.12
8	14.71	55.84	11.00
9	14.71	55.84	11.00
10	14.65	141.10	10.19

Table 13: Data Set 1 EDD rule

Data set 1 EDD rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	15.40	57.58	21.35

2	15.26	59.91	19.35
3	15.14	57.58	17.51
4	14.82	69.92	12.70
5	15.02	64.60	15.75
6	14.01	73.55	0.50
7	14.01	73.55	0.50
8	13.80	73.55	-2.73
9	13.80	73.55	-2.73
10	13.76	112.92	-3.24

Table 14: Data Set 2 MinSlack rule

Data Set 2 MinSlack Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	16,27	24,36	-10,02
2	16,27	24,36	-10,02
3	16,15	41,61	-11,60
4	16,15	41,61	-11,60
5	16,15	41,61	-11,60
6	16,15	41,61	-11,60
7	15,74	65,17	-16,86
8	15,00	74,87	-26,59
9	15,00	74,87	-26,59
10	15,00	74,87	-26,59

Table 15: Data Set 2 EDD Rule

Data Set 2 EDD Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	19,17	76,56	27,70
2	18,16	77,86	14,50
3	18,16	77,86	14,50
4	18,16	77,86	14,50
5	18,16	77,86	14,50
6	18,16	77,86	14,50
7	18,16	77,86	14,50
8	15,00	74,87	-26,59
9	15,00	74,87	-26,59
10	15,00	74,87	-26,59

Table 16: Data Set 3 MinSlack Rule

Data Set 3 MinSlack Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	15,34	29,39	-14,10
2	15,38	29,39	-13,43
3	15,19	30,62	-16,44
4	15,19	30,62	-16,44
5	15,19	30,62	-16,44
6	15,19	30,62	-16,44
7	15,05	29,39	-18,72
8	15,05	29,39	-18,72
9	15,05	29,39	-18,72
10	15,05	29,39	-18,72

Table 17: Data Set 3 EDD Rule

Data Set 3 EDD Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	15,13	29,39	-17,47
2	15,12	29,39	-17,62
3	15,12	29,39	-17,62
4	15,05	29,39	-18,72
5	15,05	29,39	-18,72
6	15,05	29,39	-18,72
7	15,05	29,39	-18,72
8	15,05	29,39	-18,72
9	15,05	29,39	-18,72
10	15,05	29,39	-18,72

Table 18: Data Set 4 MinSlack Rule

Data Set 4 MinSlack Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	16,65	46,64	11,41
2	16,66	46,64	11,49
3	16,58	46,64	10,35
4	16,37	46,64	7,17
5	16,37	46,64	7,17
6	16,36	61,58	7,03
7	16,36	61,58	7,03
8	16,36	61,58	7,03
9	16,36	61,58	7,03
10	16,28	78,38	5,84

Table 19: Data Set 4 EDD Rule

Data Set 4 EDD Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	16,52	43,35	9,41
2	16,10	63,15	3,17
3	16,10	63,15	3,17
4	16,12	54,75	2,73
5	16,12	54,75	2,73
6	15,27	70,43	-9,35
7	15,27	70,43	-9,35
8	15,22	70,43	-10,10
9	15,22	70,43	-10,10
10	15,22	70,43	-10,10

Table 20: Data Set 5 MinSlack Rule

Data Set 5 MinSlack Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	14,98	38,67	-31,87
2	14,76	38,67	-35,39
3	14,52	20,68	-39,25
4	14,52	20,68	-39,25
5	14,20	16,48	-44,25
6	13,92	8,08	-48,73
7	13,92	8,08	-48,73
8	13,92	8,08	-48,73
9	13,92	8,08	-48,73
10	13,57	8,08	-54,32

Table 21: Data Set 5 EDD Rule

Data Set 5 EDD Rule			
Value of w	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
1	14,12	0,88	-45,58
2	13,94	16,48	-48,49
3	13,90	16,48	-49,12
4	13,66	8,08	-52,97
5	13,62	8,08	-53,61
6	13,62	8,08	-53,61
7	13,62	8,08	-53,61
8	13,31	8,08	-58,55

9	13,31	8,08	-58,55
10	13,31	8,08	-58,55

Table 22: Comparisons Between w=1 and w=10

Data set 1 MinSlack rule	Average Completion Time (Hours)	Max Lateness (Hours)	Avg Lateness (Hours)
w=1	15,41	62,85	21,62
w=10	14,65	141,10	10,19
Delta	0,76	-78,25	11,43
Data set 1 EDD rule			
w=1	15,40	57,58	21,35
w=10	13,76	112,92	-3,24
Delta	1,64	-55,34	24,59
Data set 2 MinSlack rule			
w=1	16,27	24,36	-10,02
w=10	15,00	74,87	-26,59
Delta	1,27	-50,52	16,57
Data set 2 EDD rule			
w=1	19,17	76,56	27,70
w=10	15,00	74,87	-26,59
Delta	4,18	1,68	54,29
Data set 3 MinSack rule			
w=1	15,34	29,39	-14,10
w=10	15,05	29,39	-18,72
Delta	0,29	0,00	4,62
Data set 3 EDD-rule			
w=1	15,13	29,39	-17,47
w=10	15,05	29,39	-18,72
Delta	0,08	0,00	1,25
Data set 4 MinSlack rule			
w=1	16,65	46,64	11,41
w=10	16,28	78,38	5,84
Delta	0,37	-31,74	5,58
Data set 4 EDD rule			
w=1	16,52	43,35	9,41
w=10	15,22	70,43	-10,10
Delta	1,30	-27,08	19,51
Data set 5 MinSlack rule			
w=1	14,98	38,67	-31,87
w=10	13,57	8,08	-54,32
Delta	1,40	30,59	22,44
Data set 5 EDD rule			

w=1	14,12	0,88	-45,58
w=10	13,27	8,08	-59,19
Delta	0,85	-7,20	13,61

Appendix F – Efficiency of the Schedules for the Different Data Sets

Table 23: Results experiment Data Set 1

Data Set 1			
Priority rule	Average completion Time (Hour)	Max Lateness (Hour)	Average Lateness(Hour)
MinSlack	14,71	55,84	11,00
Steepest Descent	14,51	51,64	8,05
Delta	-0,20	-4,20	-2,95
%change	-1,34%	-7,52%	-26,80%
EDD	13,79	73,55	-2,73
Steepest Descent	13,48	67,60	-7,43
Delta	-0,31	-5,95	-4,69
%change	-2,27%	-8,08%	-171,79%

Table 24: Results experiment Data Set 2

Data Set 2			
Priority rule	Average completion Time (Hour)	Max Lateness (Hour)	Average Lateness(Hour)
MinSlack	16,27	24,36	-10,02
Steepest Descent	14,49	2,38	-33,19
Delta	-1,78	-21,98	-23,17
%change	-10,96%	-90,22%	-231,27%
EDD	15,00	74,87	-26,59
Steepest Descent	14,36	42,46	-34,82
Delta	-0,63	-32,41	-8,22
%change	-4,22%	-43,29%	-30,93%

Table 25: Results experiment Data Set 3

Data Set 3			
Priority rule	Average completion Time (Hour)	Max Lateness (Hour)	Average Lateness(Hour)
MinSlack	15,05	29,39	-18,72
Steepest Descent	14,72	25,19	-23,91
Delta	-0,32	-4,20	-5,19
%change	-2,15%	-14,29%	-27,71%
EDD	15,05	29,39	-18,72

Steepest Descent	14,72	25,19	-23,91
Delta	-0,32	-4,20	-5,19
%change	-2,15%	-14,29%	-27,71%

Table 26: Results experiment Data Set 4

Data Set 4			
Priority rule	Average completion Time (Hour)	Max Lateness (Hour)	Average Lateness(Hour)
MinSlack	16,37	46,64	7,17
Steepest Descent	16,01	42,44	1,73
Delta	-0,36	-4,20	-5,44
%change	-2,22%	-9,01%	-75,89%
EDD	16,13	54,75	3,48
Steepest Descent	15,76	50,55	-1,96
Delta	-0,36	-4,20	-5,44
%change	-2,25%	-7,67%	-156,41%

Table 27: Results experiment Data Set 5

Data Set 5			
Priority rule	Average completion Time (Hour)	Max Lateness (Hour)	Average Lateness(Hour)
MinSlack	13,57	8,08	-54,32
Steepest Descent	13,22	-13,88	-59,97
Delta	-0,35	-21,95	-5,65
%change	-2,60%	-271,83%	-10,40%
EDD	13,31	8,08	-58,55
Steepest Descent	12,96	-13,88	-64,20
Delta	-0,35	-21,95	-5,65
%change	-2,65%	-271,83%	-9,65%