MASTER

On-Time Delivery
Improving Planning And Scheduling In A Multi-Stage Manufacturing System

Giesen, Bas J.

Award date:
2022

Department of Industrial Engineering and Innovation Sciences
Operations, Planning, Accounting and Control Research Group

On-Time Delivery: Improving Planning And Scheduling In A Multi-Stage Manufacturing System

Bas Johannes Giesen

TU/e supervisor:
Dr. Ahmadreza Marandi
Dr. Vinh Dang
Dr. Nico Dellaert

Philips supervisor:
Niels Castien

In partial fulfilment of the requirement of the degree of Master of Science in Operations Management and Logistics

Eindhoven, June 20, 2022

TU/e, School of Industrial Engineering
Operations Management and Logistics

Keywords: anti-scatter grids; delivery performance; multi-stage manufacturing system; discrete event simulation (DES); genetic algorithm (GA); dual resource constraint (DRC); flexible job shop scheduling problem (FJSSP); scarce setup operators (SSO); capacity planning.

Abstract

In this thesis, the production planning of a multi-stage manufacturing system is optimized in order to improve the delivery performance. The manufacturing system consists of three stages and every stage is optimized in a different way. Stage 1 is optimized using a discrete event simulation (DES) to generate an inventory policy which guarantees a 99% fill rate. A Genetic Algorithm (GA) is used to tackle a dual resource constraint flexible job shop scheduling problem with scarce setup operators(DRC-FJSSP-SSO) in stage 2 and capacity planning is used to optimize stage 3. Finally, the stages are combined and the new total production time is compared to the current one.

Executive Summary

This thesis is conducted at Royal Philips NV at the 1D grids factory in Best. At this location, anti-scatter grids are produced which are used in x-ray scanners to sharpen the image and reduce the amount of radiation required. Within Philips, these anti-scatter grids are produced from raw materials. In this production process, there are several challenges, e.g. the yield is low, the capacity of machines is limited and the capacity of skilled operators is lacking. To deal with these challenges and to still have a good delivery performance, good planning of capacity and scheduling of orders is required. Therefore, the main research objective of this thesis is to provide recommendations on how planning and scheduling of anti-scatter grids, within a multi-stage manufacturing system, can improve the delivery performance to close the gap to the aimed 95%.

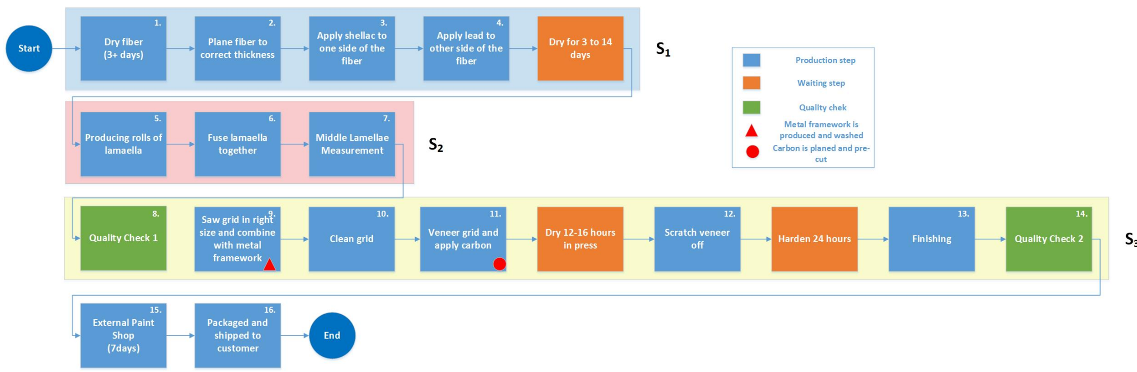


Figure 1: Process Flow for a regular grid

Figure 1, shows what the multi-stage manufacturing system looks like. At Philips, the system will consist of three stages where each of the stages will be optimized in its own way. Stage 1 will be optimized using a discrete event simulation. The goal is to generate an inventory policy that guarantees a 99% fill rate. Stage 2 can be seen as a dual recourse constrained flexible job shop scheduling problem with scarce setup operators. This is a scheduling problem where a job is produced on an eligible machine and the machine has to be set up by an eligible operator. This is a complex problem and that is the reason why exact methods will not tackle this problem in a reasonable computation time. Therefore, a genetic algorithm, based on Obimuyiwa (2020) is developed. A genetic algorithm is based on concepts of evolution and natural selection. The idea is to create random solutions for a given optimization problem and 'evolve' the solutions towards optimal solutions. The optimal solution is in this case based on the selection pressure induced by the objective function. A genetic algorithm is developed because it is used most often for these types of problems, it is very effective for combinatorial problems, it is very effective when handling large search space, it is a simple process and has strong extensibility. Using this genetic algorithm, a schedule for stage 2 can be generated. For stage 3, the amount of operators at each processing step is determined based on the number of grids that are produced per day. All activities within stage 3 are mapped with corresponding times. With this knowledge, it can be determined how many grids a processing step can handle with X operators. This means that it can be determined how many operators are required when producing several grids in stage 2.

To incorporate stage 1 into the other two stages, an assumption is made. It is assumed that, if the reorder levels and review period are used, the inventory will never drop below 0. This means, there is always enough inventory to start production in stage 2. This is a realistic assumption because the inventory policies are based on a 99% fill rate. Combining stage 2 and stage 3 is done by interpreting stage 3 as one unit. This unit, Post-processing, is then added to the GA. When the processing time is determined, the steady-state of the total production time can be determined using a Markov chain.

For stage 1, inventory policies are determined which guarantee a 99% fill rate. The inventory policies have to be updated over the months since the demand in the simulation is based on historic demand. To make sure that new demand patterns are taken into account, the demand in the discrete event simulation has to be updated.

The genetic algorithm gave the following results. When using 3 operators at Operation 2 during every shift, it is at least possible to process 150 jobs per day. This results in 750 jobs per week. When taking yield into account, this results in 488 jobs per week which are completed from the 750 jobs. 488 jobs per week are above the current target amount of jobs that have to be finished per week. If only 2 shifts are used per day, at least 98 jobs can be processed per day. This leads to 325 jobs that are completed per week if the average yield is used for the calculation. This is slightly below target and therefore, an extra shift can be useful. Next to that, the results show that the GA tries to schedule jobs with similar heights (and line types) on the same machine. This is useful to minimize setup times. When the height (and line type) differ, the machine requires a setup by an operator or maintenance operator. A height difference requires a larger setup than a difference in line type. That is why machines rarely change the height, but do change line types.

Capacity planning is used to optimize stage 3. Based on the number of grids produced in stage 2, a set of eligible operators has to be active at each processing step. For example, a scenario which is realistic and requires the fewest number of operators is: Post-processing 1 = 2-2-0, Veneer = 1-1-1 and Post-processing 2 = 1-1-1. This means that for Post-processing, 2 operators are required for shifts 1 and 2. For Veneer a single operator is required in every shift and the same goes for Post-processing 2. With these operators, 100 grids can be processed per day.

When combining all three stages, the total production time can be determined. The current total production time and the improved total production time are both shown in Figures 2 and 3 respectively. It can be seen that 90% of all grids are completed within 29.6 days when using the current scheduling method and 22.3 days when using the improved method. This improvement is purely based on having the right number of operators at the right processing step and releasing a new production order once a grid is scrapped.

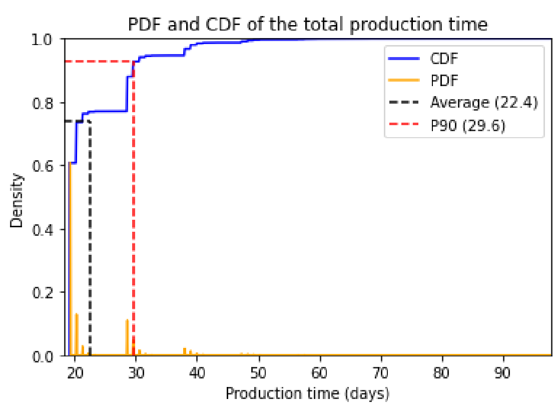


Figure 2: Current total production time

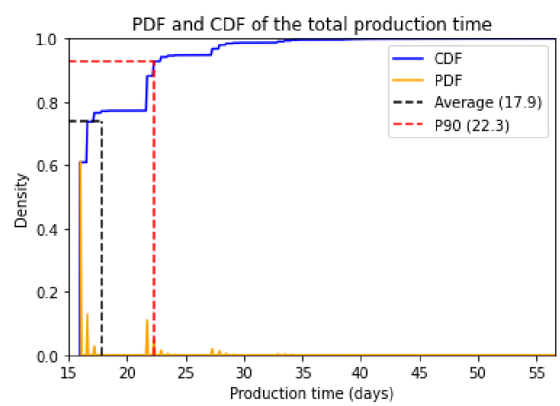


Figure 3: Improved total production time

To summarize, Philips should take new reorder levels into account to make sure that there is always enough stock to start producing grids. Next, 3 operators are required for the fusing step in the production line. The grids should be scheduled based on the characteristic "height". the setup times will be minimized when multiple grids with the same height are processed consecutively on the same machine. Finally, the number of operators at each processing step in stage 3 should be based on how many grids are produced in stage 2 of the system. The bottleneck rate of stage 3 should at least be larger than the number of grids that are produced in stage 2 minus the yield.

When keeping all these improvements in mind, the total production time will decrease and it will be possible to guarantee a 6-week delivery period. There is only one problem. Currently, there is a large backlog which means that products are not produced right away. If this takes longer than two weeks, less than 90% of the grids can be delivered within six weeks. This is because 29.6 days are required to complete 90% of the jobs. Therefore, another recommendation would be to spend a couple of weeks with the maximum number of operators in every shift to reduce this backlog.

Preface

I would like to dedicate this section to the people who supported me during the final phase of my masters.

First of all, I would like to thank everyone from Philips who helped me during this project. Especially, I would like to thank Niels and Wietse. Niels for being a great supervisor. Who helped me improve my writing skills, but also for sparring with me about more in-depth scientific methods. And Wietse for giving me all the detailed information about the production line and discuss which assumption would be best to make.

Next, I want to thank all my supervisors from the TU/e for giving me feedback and discuss challenges which presented themselves along the way. A special thanks to Ahmadreza. Although we have only met in person around three times, it still felt like you were really involved during my master thesis. It was great having you as my mentor, due to the fact that you were very involved and enthusiastic during my whole master.

Contents

List of Abbreviations

STB	Shellac Tegen Balk
LTB	Lood Tegen Balk
IC	Imaging Components
WIP	Work In Progress
OTTR	On Time To Request
OEM	Original Equipment Manufacturer
RDD	Requested Delivery Date
KPI	Key Performance Indicator
CT	Computed Tomography
IGT	Image Guided Therapy
MLM	Middle Lamellae Measurement
CR&R	Component Repair & Refurbishment
DXR	Diagnostic X-Ray
JSSP	Job-Shop Scheduling Problem
FJSSP	Flexible Job-Shop Scheduling Problem
DRC	Dual Resource Constrained
MILP	Mixed Integer Linear Program
GA	Genetic Algorithm
IP	Inventory Position
IOH	Inventory on Hand
CT	Cycle Time
TH	Throughput
DES	Discrete Event Simulation
SSO	Scarce Setup Operators
LP	Linear Programming

List of Figures

List of Tables

1 Introduction

X-ray scanners are used worldwide. The concept of these scanners is based on x-rays entering diverse parts of an object differently based on their density. A difference in density can therefore be shown in the images when the x-rays reach the detector. In the medical world, images, which are created using x-rays, can detect bones and other structures in the body like the heart and some types of tumors. In most cases, x-ray scanners are used to make a general diagnosis.

To sharpen the image, anti-scatter grids can be used. An anti-scatter grid is a grid that filters out scattered radiation, which only negatively influences the sharpness of the image. Philips produces these anti-scatter grids. The production of these anti-scatter grids is performed in the Imaging Component (IC) Factory. Philips is specialized in making fiber-interspaced anti-scatter grids. A type of paper (fiber) is used as a component of their grid. By doing this, instead of using plastic or aluminum, less radiation is required during the scan. This is because radiation passes more easily through fiber than through other materials. In this way, the anti-scatter grids not only sharpen the image, but also reduce the amount of radiation required for the scan. X-ray scanners are crucial tools for doctors, but there are risks associated with radiation. Radiation can cause damage to the cells in our body. That is why doctors want to minimize the amount of radiation that is used during a scan. Hence, fiber-interspaced anti-scatter grids are preferred.

This master thesis is performed at Royal Philips NV and is aimed to provide recommendations on how production planning and scheduling of anti-scatter grids can improve delivery performance. In this chapter, Section 1.1 will provide further information regarding the context of the project. Section 1.2 will define the problem statement and Section 1.3 will give a complete representation of the scope of this thesis. Finally, the last section of this chapter (Section 1.4) will present the research questions. In Chapter 2, the Literature background is discussed. Chapter 3 explains which methods are used in this thesis. Chapter 4 will show the results of the models and in Chapter 5, these results will be discussed and the research questions will be answered.

1.1 Context Description

Section 1.1.1 will give some in depth information about Philips. Next, Section 1.1.2 will explain in more detail about the factory where this thesis is conducted. Section 1.1.3 gives a clear overview of how anti-scatter grids work and finally, Section 1.1.4 will explain how an anti-scatter grid is produced.

1.1.1 Royal Philips NV

In 1891 Royal Philips NV was founded. The focus in that day was the carbon-filament lamp, but over the years, Philips started to produce a variety of products in numerous markets. Currently, Philips is almost every year in the top 10 patent applicants to the European Patent Office. This shows the innovative aim of Philips. Today, Philips's product portfolio has however drastically changed. Currently, Philips focuses only on healthcare-related products. " At Philips, our purpose to improve people's health and well-being through meaningful innovation is at the heart of everything we do. Never has this central tenet been more important than it is now, in these challenging times." (Philips, 2021).

1.1.2 IC Factory

This master thesis will be performed at Philips Medical Systems at Factory Best. The department within Factory Best is Component Repair & Refurbishment (CR&R). The IC Factory, within CR&R, is the area where this thesis will be conducted. In Best, fully working medical systems are assembled, ready to ship to the customer. Within the IC Factory grids are produced that are used for internal Philips departments (e.g. Image Guided Therapy (IGT), Diagnostic X-Ray (DXR), Computed Tomography (CT)) and Original Equipment Manufacturer (OEM) customers. Within the IC factory, 2 types of grids are produced. Internally, these are called: 1D and 2D grids. 1D grids are Smit Röntgen fiber-interspaced anti-scatter grids that are used for IGT- and DXR systems. 2D grids are Tungsten anti-scatter grids that are used for CT scanners. The focus of this master thesis will be to come up with recommendations for the production schedule of 1D grids. The purpose of a grid and how a grid is produced can be read in the Chapters 1.1.3 and 1.1.4.

1.1.3 Anti-Scatter Grids

An Anti-Scatter Grid is used to create sharp images using x-rays. The purpose of the grid is to filter out the scattered radiation. Scattered radiation is sometimes formed when the radiation passes through the object which has to be scanned. It bounces off something within the scanned object and is therefore no longer useful for the image and can only negatively influence the image. The grid makes sure that most scattered radiation is filtered out. This is done by using thin fiber and lead strips. These are glued to each other at an angle. Lead absorbs the radiation while fiber let the radiation pass through. This means only radiation that has not changed direction will pass through the grid and will land on the detector as can be seen in Figure 4. The blue strips are the lead strips (these absorb the scattered x-rays) and the white strips are the fiber.

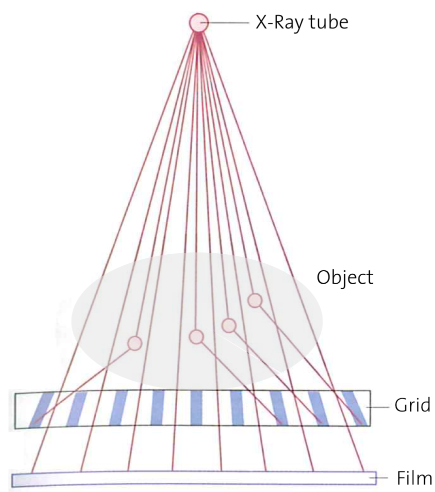


Figure 4: Anti-Scatter Grid (Cattin, 2016)

How the grids are produced and how the complete production line looks, will be described in the following chapter.

1.1.4 Process flow

As mentioned before, grids are produced from small strips of lead and fiber, called lamellae. However, the first processing step is not combining these strips. The process of producing a grid starts with: rolls of fiber, rolls of lead, shellac, glue, metal frameworks and carbon. How these materials will be combined is in detail explained below. Because certain types of grids need different processing steps only the process flow of a regular grid is explained. The flow is also visualized in Figure 5 and enlarged in Appendix A.

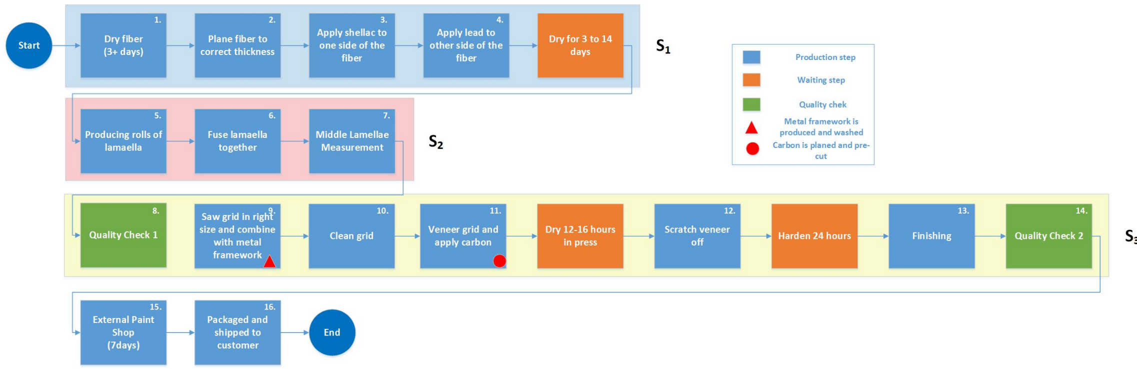


Figure 5: Process Flow of a regular grid

1. Oven

Once the rolls of fiber are received from the supplier they are stored within the production location. This is because this production line has a low humidity-controlled climate. This means that before the fiber can be used it has to dry for multiple days/weeks. To speed up this process, rolls of fiber can also be placed in an oven. They have to be in for around 3 days. In total 5 ovens can be used to dry the fiber. Once the rolls have been dried they are ready for the next step. In total 5 different types of fiber are used to produce grids. Each type of fiber differs in thickness.

2. Planing

The second step is called planing. In this step, the fiber is planed to a certain thickness. The 5 types of fiber can create a total of 13 different thicknesses of fiber. Planing is done using a machine where the roll is unrolled, planed and then rolled back up again. Every roll of fiber has to be planed at least 4 times to create the right thickness. In total 2 planing machines are used. Once the fiber has the right thickness it can go on to the next step.

3. Apply shellac

In this step, the shellac is applied to one side of the fiber. Shellac is a type of glue which is necessary for this process. This is because it dries quickly and once it is dry it can become sticky again when it is electrified. Once this step is finished the roll of fiber has turned into a roll of fiber with shellac on one side of it.

4. Cache

After the shellac is applied to one side, the other (empty) side can be glued to lead. This is done with two-component glue. A layer of lead is glued to the empty side of the roll of fiber. Once the lead is applied, the end product is a roll of fiber with on one side shellac and the other side a thin layer of lead. This is called a cached roll. Because the 2 component glue has to dry these rolls have to be placed in storage for 3 days. There is one exception. One type of cached roll has to dry for 14 days. This is because the yield will increase (in quality checks 1 and 2) if it dries for a longer time. There are three different types of lead. They also differ in thickness. The combination of the type of lead and type of fiber is pre-determined.

5. lamellae roll maker

Once the glue has dried for 3 or 14 days it is ready to go to the next stage. This is where the lamellae are created. A cached role is placed into the machine and a roll with a pre-determined amount of lamellae is produced. These rolls of lamellae are produced per order. Therefore a roll of lamellae is already linked to a specific type of grid to be produced. In total there are three machines which can perform this step. They only differ in the type of lamellae they can produce. They can produce lamellae with different heights. Once a roll of lamellae is finished it is stored in a rack. For each sticking machine, which will be discussed in the next production step, there is enough storage to hold two rolls of lamellae.

6. Sticking/fusing

In this step, the grid can be produced. The roll of lamellae is placed on a specific sticking machine. The machine glues the lamellae one by one to each other at a predetermined angle. The lamellae is placed against the previous lamellae. Next, the lamellae are electrified and the shellac becomes sticky again. Now the shellac is sticky, the lamellae is glued to the previous lamellae and stays in place. In total there are 34 sticking machines. Not all of them can produce the same type of grids. There are two types of sticking methods, e.g. lead against beam (LTB) and shellac against beam (STB), which produce different types of grids. Currently, 27 machines use the STB method and 7 machines use the LTB method. Besides this, the machines differ in other settings. Every machine can produce only a limited variety of grids. On average 8 different grids can be produced on a machine.

The production length of one grid also differs per type. For some grids, the sticking process only takes 3 hours, while for others it can be 7 hours. It depends very much on the thickness of the fiber and the size of the grid. Before machines can be used for production, they have to be cleaned and set to the correct settings. This can be seen as the setup for each production.

7. Middle Lamellae Measurement

Once the grid is produced it goes directly to the Middle Lamellae Measurement (MLM). A line is drawn on the lamellae which is completely vertical. After this step, the grid can be checked for errors.

8. Quality Check 1

Now the line is drawn at the most vertical lamellae of the grid it is ready for the first quality check. The newly created grid is placed in between the detector and the radiation tube. Next, an image is created using x-rays. The image is checked by software and a qualified operator. If the image shows errors, the grid is rejected and recycled. If the grid has passed the quality check it can go to the next step.

9. Sawing

In this processing step, the grid is sawn into the right size. Once this is completed, the internally produced metal framework is placed around the grid. This framework is produced and washed in a different part of the factory.

10. Cleaning

Now the grid has a metal framework around it the grids are sanded and cleaned.

11. Veneers

Once the grid is cleaned, it is ready to get veneered. Once the veneer has been applied, a pre-cut piece of carbon is placed on top of the grid. This carbon first has to be cut and sanded before it can be used. When the carbon is applied the grid has to dry for 12-16 hours in a press.

12. Scraping off

As soon as the 12-16 hours are over the grids can be taken out of the press. It has to be unpacked and scanned. Once this is done, parts of the veneer can get scraped off. This process is harder for the operator if the grid is left longer in the press. Once the veneer is scraped off, the grid has to harden for 24 hours.

13. Finishing

In this step, all grids are milled and engraved. It is also possible for the customer to add additional options like holes or a handle. In total there is 1 milling machine, but currently, a new machine is ordered which can mill and perform other actions twice as fast. Other machines are used for engraving and drilling. Multiple actions can be performed in this step.

14. Quality Check 2
Once all production steps within Philips have been passed, the grid is once more checked for errors. This is again done by creating an image using radiation with the grid placed in front of the detector. However, in some cases, there is a stricter quality check than the one before. This is only done at customers' requests.

15. Painting
Once the grid is finalized and approved some grids have to go to an external paint shop. Every Monday a shipment is sent and received. This step increases the total lead time by at least 1 week. Sometimes it is even more. If the product is for example ready on a Tuesday, it has to wait until Monday before it can be shipped to the paint shop.

16. Packaging & Shipping
Once the grid has returned from the paint shop it can be packed and sent to the customer or be placed in the storage location. The shipping is done by a third-party logistic provider.

1.2 Problem Statement

Providing Philips- and OEM customers with anti-scatter grids is important. It is even more important to make the Requested Delivery Date (RDD). If the safety stock at the ordering Philips department is empty and the grids are not delivered on time, shipments of complete x-ray scanners are delayed. OEM customers are likely to stop ordering at Philips once the order is delivered after the RDD. This means that they start ordering their grids from Philips' competitors, because Philips is a non-reliable supplier. This can lead to loss of sales and will negatively influence Philips.

When a customer orders one or more grids at Philips, the sales department tells the customer that the grid(s) will be delivered in 6 weeks. The On Time To Request (OTTR) is the delivery performance Key Performance Indicator (KPI). Currently, the OTTR is on average 65% as can be seen in Figure 6. This means that on average 35% of all orders are not delivered within 6 weeks. Philips aims to improve this KPI to 95%.

According to Hopp and Spearman (2011), there can be multiple aspects that negatively affect the delivery performance within a factory: (1) An unrealistic RDD. (2) Not being able to start production due to a shortage of materials. (3) Too low capacity of machines and or operators combined with a large demand. (4) The cycle time, the time it takes to do a process, can also be a factor when it becomes larger than the 6 weeks. In this case, the RDD is realistic, but the production process can be further improved to reduce cycle time. In the case of (1), the 6-week processing time might be too optimistic. All 4 aspects will be further explained in the upcoming subsections.

The RDD can be seen as a given variable during this master thesis. The current processing time is set to 6 weeks. The results of this master thesis can determine if 6 weeks is a feasible period. Aspects 2 and 3 can be influenced by having a good planning. During the production process, the capacity of machines and operators can be affected to make sure you will not try to produce more than is possible with the available resources. Good planning will contribute to fewer material shortages. Aspects 3 and 4 can be improved with scheduling. Krajewski et al. (2010) mentions that effective production scheduling is also very important for successful factories. Allocating resources over time for a certain task is necessary because otherwise the workload will be unbalanced and this can cause production uncertainty in organizations (Zhang and Wang, 2016). How these factors influence the delivery performance and how Philips currently performs is explained in the subsections below.

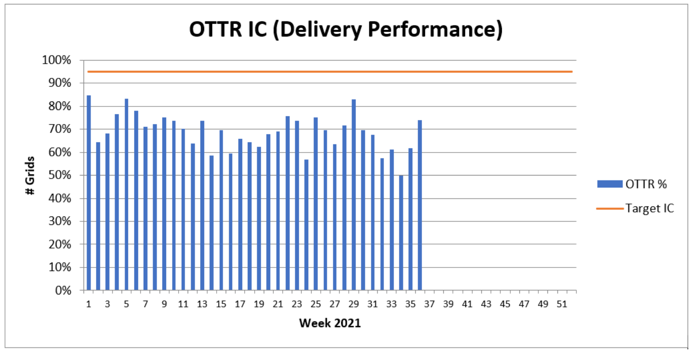


Figure 6: On Time To Request performance

Requested Delivery Date

The time in which a grid has to be produced and shipped to the customer is 6 weeks. Currently, only 65% of the orders make it to the customer in these 6 weeks. It is not clear whether these 6 weeks are an accurate reflection of the time that is required to produce and ship a grid. It might be the case that the sales department is promising unrealistic delivery dates. That means that the production process cannot be finished before it has to be delivered. To find out how well the production line is currently performing and how well it can perform, Hopp and Spearman (2011) defined simple formulas. The production lines' best-, worst- and practical worst case scenarios of the throughput and cycle time can be determined using these formulas. Once these are calculated, the actual values of the throughput and cycle time can be determined and it can be examined if the production process is performing well or not. How to determine these scenarios will be explained in Section 2.3.

Raw material Shortage

Within a production process, it can occur that the production cannot start due to a shortage of materials. This can for example be caused by a bad planning, which means that the materials are ordered too late, and mistakes from the supplier of the materials. Within the IC factory, this rarely occurs. This is because only a small number of products are required, from a local and mature supply chain, to produce anti-scatter grids. Therefore, this possible cause for the low OTTR will be kept out of scope.

Capacity

The capacity of this production line consists of two aspects, machines and operators. There are a limited amount of machines available and some of them also have a fixed configuration which means that they are not able to produce all types of grids. Operators control the machines. Most operators are trained in one or multiple processing steps. This means that they can perform these steps, but are not qualified to work on processing steps somewhere else down the production line. This means that if the whole production line is required to be active, the operators with the right training are necessary. Every production line has a maximum capacity of products it can produce per time unit. If demand is increasing and more products have to be produced, the machine may reach its maximum capacity. This means capacity (machines and/or operators) has to increase, fewer orders have to be accepted by the sales department, or the production process has to be improved which will increase the throughput of the production line. In the following sections, machine- and operator capacity and growing demand are explained in more detail.

Machines

The first capacity restriction is regarding the number of machines and their configuration. For the sticking machines, the constraint is the configuration of the machines. Not all grids can be produced on every machine. This can lead to machines being not active. Besides that, these machines are also very sensitive to failures. This means that sometimes a machine is down and maintenance has to take place. Also, there is only one milling machine. On this machine, all grids have to be processed and a large part of the frameworks have to be produced. The combination of processing grids and producing frameworks means that the machine almost is a utilization runner to fulfill demand. That is why a new milling machine is ordered which has twice the capacity. Besides that this machine can also directly add engravings and holes to the grid, which saves extra processing steps.

Operators
As explained before, every operator specializes in different processing steps. For example, there are only 3 operators which are qualified/trained to perform quality checks. This directly means that if 2 operators are for some reason absent, fewer grids can be checked that day and this leads to a large inventory. Next to that experienced employees work faster than others. This means that for example, 2 experienced operators can almost activate all ”sticking machines”, while this is for 3 regular operators also the limit. The output of a processing step depends very much on which operator(s) is/are working. To conclude, the lack of employees, with the right skill set, ensures that certain processing steps could not perform as well as planned. This can lead to a longer cycle time and it can increase the Work In Progress (WIP) and decrease the OTTR.

Demand
As can be seen in Figure 7 the sales forecast of 1D grids (for the upcoming 6 weeks) plus backlog has kept increasing since the start of 2021. It has never been this high before. Besides that, the 2D grids, which are also produced in the IC Factory (but on other machines), have an even larger demand growth (not shown in the Figure). During the past years, more time went into the production and planning of 2D grids. This led to less attention to the 1D grid production process. It is, therefore, less improved over the past couple of years than desired, while demand has significantly grown.

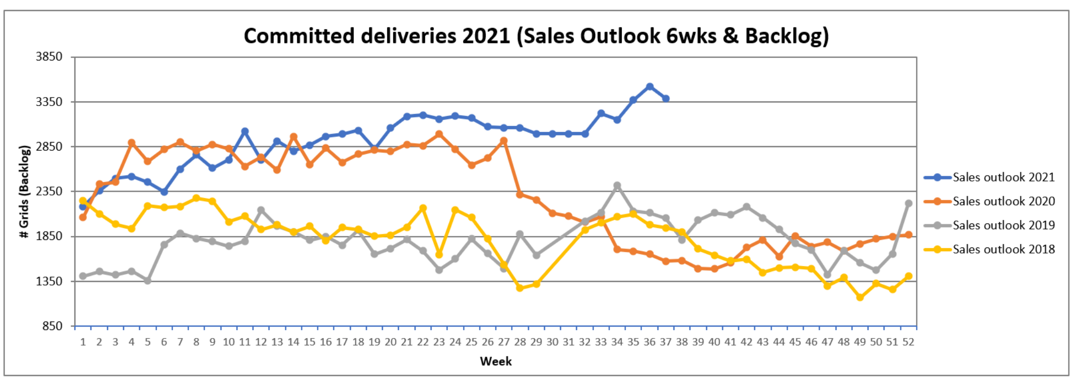


Figure 7: Sales forecast upcoming 6 weeks

Cycle Time

Cycle time is the sum of activity durations, fewer overlaps between activities, plus the sum of queue times, consequently, cycle times can be reduced by reducing queue times, overlapping activities and reducing activity duration (Ballard, 2001). A long job cycle time results in the accumulation of WIP (Hopp and Spearman, 2011). Besides that, the risk that the product is polluted increases if the cycle time is long (Bon and Samsudin, 2018). According to Hopp and Spearman (2011), a larger cycle time leads to less flexibility, with respect to releasing the orders and lower quality of the products, because products are placed for a longer time in a queue. There are different reasons which influence the cycle time according to Hopp and Spearman (2011).

WIP

The WIP is directly related to the cycle time and the throughput. This relationship is defined as Little's Law (Little, 1961). L defines the WIP, λ is the throughput of the machine/system and W is the cycle time.

$$L = \lambda \cdot W \tag{1}$$

From Equation 1 it can be concluded that a high WIP leads to a high Cycle Time if the throughput is kept constant. The variability buffering law (Hopp and Spearman, 2011), implies furthermore that if you want to reduce the WIP but do not reduce the variability, it will cause the throughput to decrease (Reyes et al., 2017).

Currently, the WIP of the anti-scatter grid production line is approximately 1700 units. Philips aims to have the WIP at 1000 units. This target is currently based on experience, so it might be the case that the aim of 1000 units of WIP is not optimizing the cycle time. A large WIP usually has some disadvantages, as explained above. It can increase the cycle time of products. This is because queues are formed before processing steps. A grid has to wait at a machine until all products, which arrived earlier, are processed. This means, that instead of going directly from step to step in a production line, the grid has to wait at every step before it can be processed.
Next to that, there will also come more opportunities for errors in the production process. Due to the large piles of inventory before every processing step grids are stationed for a longer time in one place. This can lead to pollution of the grids which decreases the yield even more. The employee responsible for the quality during the production process is also convinced this is currently the case. Prioritizing orders, when there is a large inventory, also becomes quite challenging. This is because large amounts of grids are everywhere and the operator has to find the grids which have the most priority.

Machine Utilization

Some people expect that to produce as many products as possible machines have to be active 24/7. This is however not the case. According to the law of utilization of Hopp and Spearman (2011), it is stated that if a station increases utilization without making any other changes, average WIP and cycle time will increase in a highly nonlinear fashion. This is because a small error within the machine can lead to consequences for all the to-be-produced products. It is currently unclear what the utilization is of machines at Philips.

Yield

Scrapping products during the production process has a negative influence on three aspects, Throughput, WIP and cycle time (Hopp and Spearman, 2011). The throughput is affected in a way that if the scrap rate is large enough, it can cause a capacity problem at a machine. Because more and more products have to pass this machine and there is a possibility that the machine does not have enough capacity to produce the grids. If machines have to be used more often due to the low yield, the variability of the line also increases, which will therefore require a larger WIP (and cycle time) to remain at a certain throughput. The low yield also affects the cycle time directly. Due to the high scrap rate, a product has to be produced more often and variability increases. Together with the fact that extra WIP is in the line which leads to longer average cycle times, and the increase of variability of cycle times, the delivery performance becomes worse. It is stated that the later the product is scrapped in the line, the worse the consequences will be(Hopp and Spearman, 2011).

Within the IC factory, scrapping occurs. As mentioned in Section 1.1.4, there are 2 quality checks within the production line. The first one is performed right after the MLM. The average yield of the past month for this check is 78.8%. This means that 21.2% of all grids are rejected due to error(s) during the quality check. The second quality check is performed once the grid is almost finished. Although you would expect that the yield should be large, because of the fact that already a quality check has been performed, the average yield for the past month is just 77.0% for this check. This means that once the grid is nearly completed 23.0% of all grids are still rejected. This has a

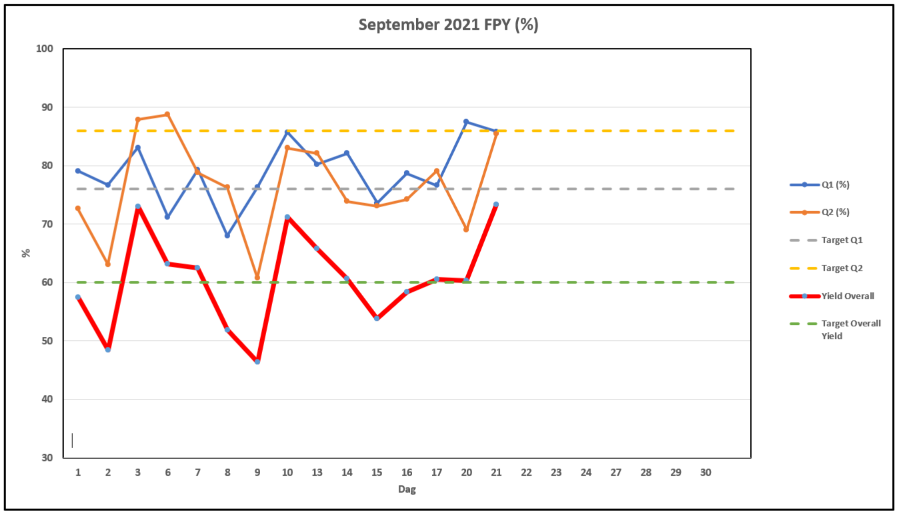


Figure 8: Yield during September 2021

greater impact than the previous quality check, because more resources went into the grid and the production of a new grid has to start all over again, while it was expected that a grid was almost complete. Combining the yield of both quality checks results in an average yield of 60.6%. This means that if 1000 grids are produced, 787 pass the first quality check and finally only 606 will be completed to ship to the customer. If the demand for the next couple of weeks is 1000 grids and we assume the yield to be 60.6%, 1650 grids have to be produced to meet the demand. For some types, the total yield is even worse than 60.6%. The overall average yield ranges from 30% to 66% per type of grid. The yield of the past month can be seen in Figure 8

Planning & Scheduling

A problem regarding the planning is that currently, the outflow of the production line is not constant and reliable. It is unclear how much materials you should use as input and when to start producing to get the ordered products at the pre-determined date. This makes it very hard to plan the production process.

Since on average 35% of the grids are rejected and therefore on average 35% of the grids have to be reproduced, the average cycle time of grids will also increase. The planning system which is used in the factory is not built to compensate for the expected loss of grids. This means that if there is an order of 20 grids, the shift leaders can only start producing 20, while it is expected when assuming the average yield, that 33 grids have to be produced to end up with 20 grids. Once 1 of the 20 grids is rejected, the shift lead has to start the production of a new grid. This, therefore, increases the lead time by a couple of times the actual lead time. This will finally affect the delivery performance.

Next to that it currently is not clear which orders have to be produced per day. The shift leaders receive a priority list which is based on the sales order list. However, the importance of each order is not shown. This means, that if the order has to be produced quickly, because the customer cannot wait any longer, the only way to 'prioritize' this order is by asking the shift leads to start the production of this grid. Currently, they get requests from multiple different employees and this gets confusing for team leads, because different employees mention that different orders have priority. This makes scheduling hard. Next, most of the time several order receipts are created per machine and given to the operators. There is no order in how these sales orders should be produced.

Conclusion

Producing the right and enough grids within the IC factory is of high importance for Philips. By doing this, the target for the OTTR KPI can be met. This needs to happen because customers expect a reliable supplier of anti-scatter grids. Besides that, Philips is not the only supplier of grids. This means that if Philips is not delivering the grids before the RDD, customers might stop ordering and go to Philips' competitors. This will lead to a significant amount of lost sales and will therefore decrease the revenue generated by this department. Meeting the OTTR target is hard for Philips. The reason why Philips struggles with the delivery performance can be caused by multiple aspects. If the delivery performance is not met, it means that 6 weeks of production time is too optimistic or that the cycle time for a grid is too high. It can also occur that raw materials are not present when they are required (but it is assumed to be not the case in this thesis) or that demand exceeds the maximum capacity of produced products. The high cycle time can be caused by different aspects. Firstly, part of the produced grids is scrapped. Currently, around 60% of all produced grids are rejected due to production errors. Secondly, the WIP within the production line is very high, which leads to large waiting times and more pollution of the grids. Thirdly the capacity of operators combined with the increasing demand can lead to the fact that the maximum capacity of the production line is reached. The demand for anti-scatter imaging grids has kept increasing, while capacity planning has almost not improved. Lastly, the priority scheduling of grids, which grid has to be produced first because it has priority, is almost not implemented in the current system. Next to that, it has hard to schedule when to start producing if the output of each processing step is irregular.
The main objective of this master thesis will be to develop a scheduling system that determines what orders to schedule in what sequence to improve the OTTR level, while keeping in mind the constraints that are present in the IC factory.

1.3 Scope

Within this thesis, multiple aspects are left out of scope. The first one is the focus on 1D grids instead of 1D and 2D grids in the IC factory. This is because the production line of both types of grids differs too much to combine them in one model. When deriving recommendations, it is tried to generalize recommendations for the 1D grid production line to determine if these are also applicable for the 2D grid production line. Next up it is assumed that there are three main types of 1D grids. Within the IC factory in total, 173 different types of 1D grids are ordered over the past year. Grids differ in size, shape and thickness. During the finishing processing steps, there is also variation possible concerning the looks of the grid. Extra holes, grips, etc. can be applied to the grid to meet the customers' requests. In total 17000 good quality grids have been produced over the past year. The grid which is most frequently produced during the past year is produced 1792 times. In this thesis, it is assumed that all grids are assigned to three main groups that are produced, e.g. Mammography-, Regular- and Round grids. Almost all grids can be subdivided into these groups. This is therefore the reason that this assumption is made. Another assumption that is made is regarding the supply of materials. As mentioned before it is assumed that there are always materials on stock to start producing grids. This can be assumed due to the mature and local supply chain. The most important scope of this thesis is that only planning and scheduling will be revised and developed. The actual production process and the associated yield are given and cannot be changed.

1.4 Research Questions

This thesis focuses on a multi-stage manufacturing system within the IC factory at Philips. Within this factory, anti-scatter grids are produced. Currently, the delivery performance is about 30 percentage points below target. To improve the delivery performance, planning of capacity (machines and human workload) and scheduling of orders have to be performed.

The main research objective for this master thesis is to provide recommendations on how planning and scheduling of anti-scatter grids, within a multi-stage manufacturing system, can improve the delivery performance to close the gap to the aimed 95%. This means that the goal is not to meet the 95% OTTR, but to improve the planning and scheduling to optimize the OTTR. The following research questions will help to meet this objective.
The main research question in this thesis is formulated as follows:

How should Philips plan capacity and schedule orders to improve the OTTR, while keeping in mind the constraints?

To help to answer the main research question, several sub research questions are formulated. The first research question is used to optimize the inventory of stage 1 of the production process.

1. *How should the inventory level of S_1 be managed to make sure further processing steps can start immediately?*

The second sub research question tries to find methods that will lead to an increase in output of stage 2 and improve reliability of this output.

2. *How can the DRC Flexible Job Shop with setup operators of S_2 be scheduled to achieve a larger and more reliable output?*

Subsequently, the third sub research question is used to answer how stage 3 of the process can have a constant flow of products based on the amount of operators that are available.

3. *How to determine the required number of eligible operators in S_3 to have a constant flow of products?*

Next, sub research question 4 will combine the results of all three stages.

4. *How can all three models for S_1, S_2 & S_3 be combined?*

The last sub research question will provide recommendations on how to improve the OTTR performance.

5. *Which changes could potentially improve the OTTR performance and how can these be implemented?*

Finally, the main research question can be answered by combining the results of the sub research questions. The answer to the main research question will give Philips insight in how to better plan capacity and schedule orders to improve the OTTR.

2 Literature Background

The production line of Philips can be split up into 3 different stages (S_1, S_2 and S_3), see Appendix A . S_1 differs the most when comparing the three stages. In the first stage, the rolls are produced to be ready for further processing steps. In stages 2 and 3, every order is produced individually. The second and third stages can again be split into 2 different parts. In S_2, processing times are larger than in the S_3 and the same route is followed for almost all different types of grids in S_2. S_3 has multiple processing steps which only take a couple of minutes to complete. It is therefore not useful to create a schedule for this part of the production line. Also, grids take a very different routing which makes scheduling even harder.

The first part of the production process is similar to an Inventory Control Model. The amount of stock is currently monitored by experienced operators. If the operator thinks there is a shortage of a certain type of product (s)he starts producing it. If the shift lead gets notified that there is a large demand for a certain type of product, the operator gets alerted and (s)he also starts producing more of these types of products. In most cases, this is going smoothly, but sometimes products are not in stock when they are required for further processing steps. Currently, this stage of the process is very much dependent on the experience of operators.

The second part of the production process is a Job Shop where every order is individually manufactured. Every order has a similar routing through the production line.

Within the last part of the system, orders have different routings and have to pass different machines/ processing steps. There are processing steps that every order has to pass, but there are also steps where only a few of the orders are handled. Besides that, processing times take only a couple of minutes which makes it hard to create a schedule.

For these three stages, different optimization methods will be explained. Section 2.1 will clarify the literature background of inventory control. In Section 2.2, the literature for optimizing stage 2 will be illustrated. Finally, Section 2.3 will define the theoretical background of stage 3 of the production process.

2.1 Stage 1 - Inventory Control

In this section, first, inventory control policies are discussed in Section 2.1.1. Afterwards, also demand forecasting is explained in Section 2.1.2.

2.1.1 Inventory Control Policies

A lot of research is done concerning inventory control(Silver et al., 1998; Cachon and Terwiesch, 2008; Nahmias and Cheng, 2009; Taylor et al., 2013; Anderson et al., 2018). Different notations are used within the performed research, but in this thesis, the notation of Silver et al. (1998) will be used. This is because Silver et al. (1998) introduced a very large classification of inventory control systems.

Silver et al. (1998) make use of two types of reviewing the inventory. Continuous and in fixed periods. In practice, a periodic review period is applied in most situations(van Donselaar and Broekmeulen, 2017). The review period is called the time between two moments when the inventory levels are reviewed and are denoted by Silver et al. (1998) with the capital letter R. Another aspect that differs in inventory policies is the replenishment quantity. In some policies, the inventory position (IP) is replenished using (integer multiples of) a fixed quantity. This is the case when ordering or production is done in batches. To denote the fixed base replenishment quantity, the capital letter Q is used by Silver et al. (1998). In other policies the replenishment quantity is variable. In these cases, the IP is replenished to a fixed order-up-to level denoted by Silver et al. (1998) as capital letter S. Most policies make use of a reorder level, denote by the small letter s (Silver et al., 1998). If the IP drops below this level a new order is placed or production is started.

Silver et al. (1998) defined multiple inventory control policies, which can be used to optimize an objective like backorders, fill rate, or inventory on hand (IOH). 5 common used policies are: (R, s, nQ), (R, s, S), (R, S), (s, nQ) and (s, S). Below, all policies are explained in detail. In the case of this thesis ordering products can be assumed to be similar to producing products.

Table 1: Multi-Period Inventory control policies

Policy	**Explanation**
(R, s, nQ)	The stock is reviewed every R periods. When the stock drops below the reorder level, s, nQ units are ordered to bring the stock back up to the reorder level.
(R, s, S)	The stock is reviewed every R periods. When the stock drops below the reorder level, s, A variable amount of units is ordered to bring the stock back up to the order up-to level, S.
(R, S)	The stock is reviewed every R periods and every period X units are ordered to bring the stock back up to the order up-to level, S.
(s, nQ)	The stock is continuously reviewed. When the stock drops below the reorder level, s, nQ units are ordered to bring the stock back up to the reorder level.
(s, S)	The stock is continuously reviewed. When the stock drops below the reorder level, s, A variable amount of units is ordered to bring the stock back up to the order up-to level, S.

To determine which policy suits a production line best, two questions have to be answered. First, does the factory work with advanced technology which keeps track of stock? Second, how many products are produced at once?

Singha et al. (2017) mentions that continuous reviewing is useless without the technology to support this. If the technology is not available, it means that an employee has to check manually what the stock level is. Sani and Kingsman (1997) for example mentions that the (s, S) policy is performing best for the management of items of low and intermittent demand. However, this policy cannot be used if technology is not available in this factory. This means, first the factory should see if the policy is possible to implement. The second question is important, because some production lines might only be able to produce batches of products, while other production lines can produce one product at a time. For Philips, it is the case that technology to continuously review the stock is not available. Besides that, they only produce in batches of 9 rolls. This means that there is only one policy applicable for the production line, e.g. (R, s, nQ).

For these policies, exact formulas can be derived to calculate for example the expected backorders, fill rate, IOH and others. To derive these formulas some assumptions are made which are, when used in practice, sometimes not realistic assumptions. Demand is for example stationary and lead times are deterministic. To relax the assumptions for demand and lead times, van Donselaar and Broekmeulen (2017) created lecture notes. Within these lecture notes, several assumptions are relaxed. This means that non-stationary demand and stochastic lead times can be implemented, while still being able to derive exact formulas.

To improve the first production segment of Philips, this production segment should have stock for all types of products when they are required. By doing this there is no delay for further processing steps. The KPI to be improved is called the fill rate. It is defined as the long-term fraction of demand delivered immediately from stock (van Donselaar and Broekmeulen, 2017). If the fill rate is 100% it means that in the long run, all demand is delivered out of stock and this means all further processing steps can start without any delay.

A disadvantage is that every product has its own inventory. This means 13 different reorder levels have to be determined for Philips. The results of these calculations have to be combined to check if these are feasible in the actual factory. This means the results of 13 different inventory models have to be combined to get the desired schedule which takes production- and storage capacity into account. Simple heuristics can be used to tackle this problem (Greeff and Ghoshal, 2004). Examples of simple heuristics are earliest due date, random, prioritizing the orders, etc.. van Donselaar and Broekmeulen (2017) also developed a prioritizing method. If a maximum amount of money is available to hold inventory it is decided based on backorder costs and fill rate which items should be kept as inventory. The input of this final heuristic will be the time between orders. The output will be a realistic schedule of production, while taking capacity into account.

2.1.2 Demand Forecasting

To create an inventory control model, future demand has to be known. This can be done by forecasting demand. Increasing demand can be forecasted using Exponential Smoothing with a Linear Trend, also known as Double Exponential Smoothing (Hopp and Spearman, 2011). A Linear trend is added, because demand for anti-scatter grids is increasing over time.

Double Exponential Smoothing is chosen, because firstly, it suits non-stationary demand, which is the case at Philips. Secondly, it is a flexible method in the sense that it updates its estimate of the trend. This property ensures that the forecasts react to changes in the trend, which is very practical in reality as the trend in demand is rarely stable over a long time period. At last, double exponential smoothing is very explainable. The user can easily understand how and why a forecast is generated. This can for example help when debugging the model.

Double Exponential Smoothing makes a forecast for the next period based on a smoothed demand estimate and a smoothed trend estimate. The demand estimate is based on the most recent demand observation and the most recent forecast. Put formally (Hopp and Spearman, 2011):

$$\begin{aligned} F(t) &= \alpha D(t) + (1-\alpha)[F(t-1) + T(t-1)], \\ T(t) &= \beta[F(t) - F(t-1)] + (1-\beta)T(t-1), \\ f(t+\tau) &= F(t) + \tau T(t), \end{aligned} \tag{2}$$

where $F(t)$ is the demand estimate at time t, $D(t)$ the demand realisation at time t, $T(t)$ the trend estimate at time t and $f(t+\tau)$ is the forecast for τ periods in the future. α and β are smoothing constants to be chosen by the user.

If seasonality is visible in the demand pattern, triple exponential smoothing can be used. This is done using Winters method shown below (Winters, 1960).

$$\begin{aligned} F(t) &= \alpha \tfrac{D(t)}{c(t-N)} + (1-\alpha)[F(t-1) + T(t-1)], \\ T(t) &= \beta[F(t) - F(t-1)] + (1-\beta)T(t-1), \\ c(t) &= \gamma \tfrac{D(t)}{F(t)} + (1-\gamma)c(t-N), \\ f(t+\tau) &= [F(t) + \tau T(t)]c(t+\tau-N), \qquad t+\tau = N+1, \ldots, 2N. \end{aligned} \tag{3}$$

Some (parts) of the equations are similar to the double exponential smoothing method. Currently the basic idea is to estimate a multiplicative seasonality factor $c(t)$, $t = 1, 2, \ldots$, where $c(t)$ represents the ratio of demand during period t to the average demand during the season. Therefore, if there are N periods in the season (for example, $N = 12$ if periods are months and the season is

1 year), then the sum of the $c(t)$ factors over the season will always be equal to N. γ is an extra smoothing constant that has to be chosen by the user.

2.2 Stage 2 - Job Shop Scheduling

Job shop scheduling or the job-shop scheduling problem (JSSP) is an optimization problem in which various manufacturing jobs are assigned to machines at particular times while trying to minimize the makespan. This is the total duration until all jobs are processed. Scheduling has direct impacts on the production efficiency and costs of a manufacturing system (Zhang et al., 2019). Since the late 50s it has attracted great research attention (Smith et al., 1956; Wagner, 1959; Bowman, 1959; Manne, 1960). Lenstra et al. (1977) showed that the JSSP is a NP-hard problem. Scheduling algorithms can be solved exactly or using approximate methods. Past decades approximate methods gained more attention, but this will be explained later on.

Over the years the JSSP has been studied in more detail and with more constraints. Deterministic JSSP(Carlier and Pinson, 1989) (Applegate and Cook, 1991), Robust JSSP (Jamili, 2016), with few operators JSSP (Paksi and Ma'ruf, 2016), flexible JSSP (Pezzella et al., 2008) and many more variants of the JSSP have been studied. In this thesis, the focus will be on a flexible job shop with a dual resource constraint. A flexible job shop or the flexible job-shop scheduling problem (FJSSP) is an extension of the classical JSSP that allows an operation to be processed by any machine from a given set of alternative machines (Chaudhry and Khan, 2016). When capacity constraints can be caused by both machines and human operators, systems are known as Dual Resource Constrained (DRC).To be more precise, it can be defined as when operators are the constraining resource who can transfer across various workstations as required (Treleven, 1989; Hottenstein and Bowman, 1998). DRC schedules however assume that operators are busy during the processing time of a product at a machine. That is why Obimuyiwa and Defersha (2020) changed this assumption to: Operators are only busy during the setup time for a product on a machine. Philips has multiple similar machines for production processes and has a limited amount of operators who are only required for the setup of a machine. That is why an FJSSP with DRC with setup operators is used in this thesis.

In Section 2.2.1, the Dual-Resource Constrained Flexible Job Shop Scheduling Problem is further explained. Section 2.2.2 clarifies the usage of a Mixed Integer Linear Programming (MILP) for this problem and Section 2.2.2 denotes multiple ways of solving this type of problem.

2.2.1 Dual-Resource Constrained Flexible Job Shop Scheduling Problem

One of the main objectives of job shop research is to align capacity and workload. Most literature assumes that capacity is the single variable (Thürer, 2018). But, in practice, most manufacturing systems are not only limited to machine capacity. They are also constrained by operator capacity (Bokhorst and Gaalman, 2009). This is because a machine requires in most cases an operator to function. Three elements have to be aligned in this type of scheduling problem: (1) the workload (or demand), (2) the machine capacity/capability and (3) the worker capacity/capability (Thürer, 2018). The main objective of operator-workload scheduling in DRC systems is to have the right amount of eligible operators to perform the required tasks at the right time (Obimuyiwa, 2020). The DRC system will perform more efficiently when operators have a diverse skill set (Azizi et al., 2010). However, according to (Gel et al., 2007), having a team with fully-trained operators is also not desired due to the high costs of training.

As mentioned above, the classical job shop scheduling problem is NP-hard (Lenstra et al., 1977). DRC systems are even more complex due to the extra resource-constrained (Xu et al., 2011). Within these systems, it is necessary to also take into account the worker assignment, where to assign a worker and how to do so. Because these additional aspects are added, analytical solutions are no longer feasible or adequate (Xu et al., 2011). Currently, other approaches such as meta-heuristics like genetic algorithms (GA) (Paksi and Ma'ruf, 2016) and Ant Colony Optimization are used to tackle the DRC type scheduling problems. The systematic review of the literature by Thürer (2018) confirms this. Currently, there is a tendency towards advanced scheduling mechanisms. Before, the DRC scheduling problem was mainly simulated, but literature tends toward mathematical modeling.

Obimuyiwa and Defersha (2020) performed a literature review of algorithms that are used for solving the DRC JSSP and DRC FJSSP. Only literature between 1997 and 2020 is reviewed. As can be seen in Figure 9, a single meta-heuristic is used most often in literature. This is a GA and it will be explained in Section 2.2.2. Next to determining the frequently used algorithms in the recent literature, Obimuyiwa and Defersha (2020) discovered a research gap. None of the revised research considers scarce resources, especially in the area of operators. For example, what happens if there are only a few operators which have to perform a setup operation on all machines, or if there is only one operators eligible to perform a setup on a certain machine. Obimuyiwa (2020) developed a model where eligible operators are used to setting up machines. How this model works will be explained in Section 2.2.2.

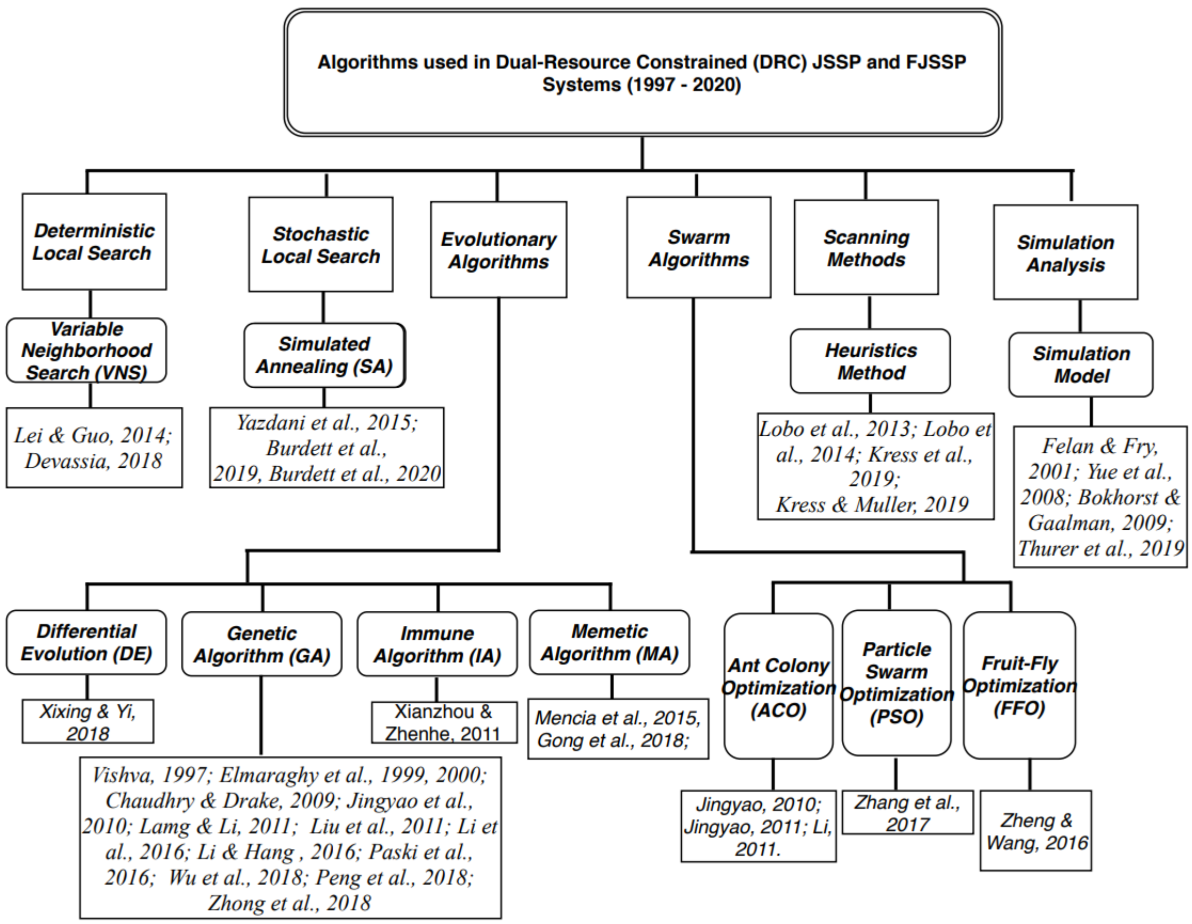


Figure 9: Classification of algorithms used in addressing DRC JSSP and DRC FJSSP (Obimuyiwa, 2020)

2.2.2 Methods

Figure 9 shows frequently used algorithms to tackle DRC FJSSP and DRC JSSP. An advantage of simulation-based scheduling is the ease of explaining the model, because the manufacturing system is modeled on the computer. It is very intuitive, because the simulation tries to copy the behavior of the real manufacturing system. By doing this more people understand what is going on and managers can be convinced with more ease. Besides that simulation is a powerful tool, dispatching rules and system design can also easily be changed (Mourtzis, 2020). This means it is easy to compare different approaches. A disadvantage of simulations is that there is no general understanding of when a dispatching rule is performing well. This means that to find a successful schedule a lot of trial-and-error has to take place (Kulkarni and Venkateswaran, 2015). Within the past decades, the DRC job-shop scheduling is mainly tackled by simulation algorithms(Treleven and Elvers, 1985). But the problem is that it fails to achieve sufficiently accurate results, despite handling simple job-shop scheduling problems (Yinan et al., 2014) (Tang et al., 2016).

Due to the complexity of the problem, meta-heuristics are more suitable (Paksi and Ma'ruf, 2016). The GA is the most frequently used method to solve this problem as could be seen in Figure 9. Pezzella et al. (2008) mentions that GAs have been successfully adopted to solve the FJSSP. GAs is used due to its practical implementation in industry (Paksi and Ma'ruf, 2016). Besides that GAs are very effective at performing a global search for combinatorial problems (Chaudhry and Drake, 2009; Werner, 2013). Obimuyiwa (2020) mentions that GAs have good accuracy in solving large-scale scheduling problems. Some studies mention that GAs are faster concerning computation time, but this is very problem-specific (Díaz et al., 2020). A GA is most often used for DRC-FJSSP, due to the fact that it has rapid random search ability, strong robustness, simple process and strong extensibility (Cellura et al., 2011; Mohammed et al., 2017; Abo-Zahhad et al., 2014). Moreover, it has been proven that it is one of the most effective evolutionary technique for solving different types of JSSP (Wu et al., 2018). Of course, there are also some disadvantages of GAs. Optimization results sometimes depend on the quality of the starting population (Zhong et al., 2018). This means that it is important how to determine the initial population.

Particle Swarm Optimization is also a frequently used meta-heuristic. Particle Swarm Optimization was first proposed by Eberhart and Kennedy (1995) for continuous optimization problems. The main advantage of Particle Swarm Optimization is that it has fewer control parameters in continuous space(Katherasan et al., 2014) (Coello et al., 2004). On the other hand, Particle Swarm Optimization is limited to combinatorial optimization problems. This is because updating the position is carried out in continuous space (Zhang et al., 2017).

Ant Colony Optimization is also frequently used since it can easily avoid additional calculation time, which can be caused by infeasible solutions (Li et al., 2011). This is because the solution is constructed by performing actions in different stages.

Another frequently used meta-heuristic is Simulated Annealing. It was first proposed by Kirkpatrick et al. (1983). It has been applied successfully for combinatorial optimization problems over the past years (Yazdani et al., 2015). Besides that, Simulated Annealing is capable of escaping local optima by allowing moves to previous solutions while searching for the global optimum (Yazdani et al., 2015).

In this section, a MILP and a GA will be explained in more detail. A MILP is discussed, because optimal solutions can be determined using this method. A GA is explained, because this is the most common method to solve this type of problem.

MILP

A common method to formulate a JSSP is by means of MILP problems. MILP is an optimization problem in which a nonempty subset of integer variables (unknowns) and a subset of real-valued (continuous) variables exist, the constraints are all linear equations or inequalities, and the objective is a linear function to be optimized (Wolsey, 2007).

Although MILP can give optimal solutions for small and medium-sized problems, the performance deteriorates with problem size. The number of decision variables in the MILP increases at a much higher rate with the increasing size of the job shop. For larger problems, the MILP model may take several hours to converge, if at all (Kulkarni and Venkateswaran, 2015). According to Ku and Beck (2016), modern MILP solvers can prove optimality for moderate-sized problems very quickly.

Different methods have been developed to solve a form of JSSP. Carlier and Pinson (1989) and Applegate and Cook (1991) have for example developed exact algorithms using MILP to solve the Deterministic JSSP with the help of Branch & Bound procedures. Past decade, also meta-heuristic approaches like GA (Spanos et al., 2014), Particle Swarm Optimization (Sha and Lin, 2010) and tabu search (Zhang et al., 2008) have been developed to solve the problem. The advantage of meta-heuristics is that it can handle larger-sized problems. Meta-heuristics usually take less time than algorithmic methods to come up with a 'good' solution for larger problems. However, they do not guarantee optimality.

Genetic Algorithm

GAs were introduced by Holland et al. (1975). A GA is based on concepts of evolution and natural selection. The idea is to create random solutions for a given optimization problem and 'evolve' the solutions towards optimal solutions. The optimal solution is in this case based on the selection pressure induced by the objective function. A GA can be very effective when handling large search spaces (Nobile, 2021). How GAs exactly work is explained below and visualized in Figure 10.

1. N individuals are created: the population
2. The fitness for all individuals is determined
3. A selection mechanism is used to create pairs of individuals (with a probability proportional to their fitness value)
4. The selected pairs exchange 'chromosomes' (part of the encoded schedule in this case) to form new individuals
5. The new individuals mutate (part of the matrix is randomly changed)
6. The new individuals (offspring) replace the old population
7. If the termination criterion is met, the solution will be the best fitting individual. Else, go to step 2

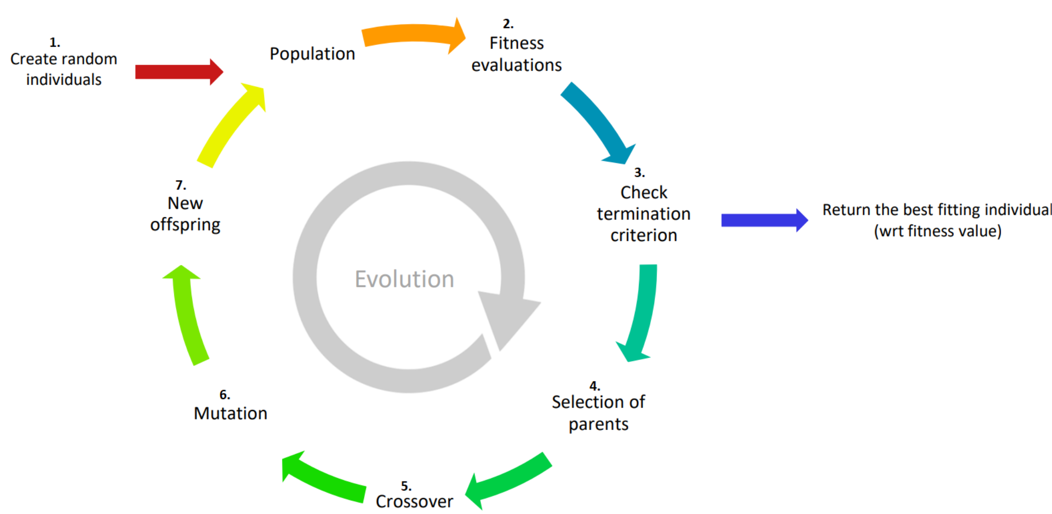


Figure 10: Genetic Algorithms iterative process (Nobile, 2021)

2.3 Stage 3 - Capacity Planning

To determine how well a production line or machine is performing, Hopp and Spearman (2011) developed some simple formulas, based on Little's Law (Little, 1961) as can be seen in equation (4).

$$L = \lambda \cdot W \tag{4}$$

where L is used for the cycle time, λ defines the throughput and W the current work in progress. With these formulas the cycle time and throughput of different scenarios can be calculated over a varying amount of WIP.

In total, Hopp and Spearman (2011) defined three scenarios, i.e. best case, worst case and practical worst case. The actual cycle time (CT) and throughput (TH) can be benchmarked against these three scenarios. In Figure 11 an example is shown how this would look like. When calculating the actual throughput and cycle time, it is desired to be in the 'good region' as can be seen in Figure 11.

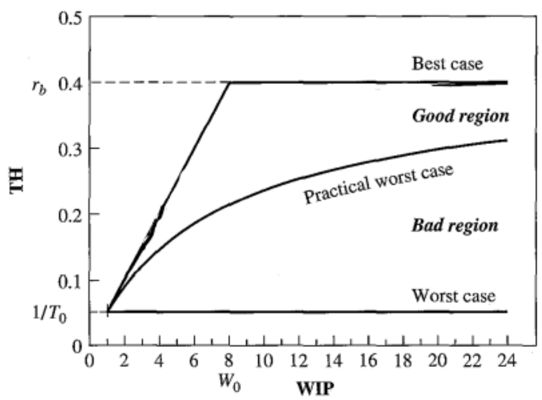


Figure 11: Example of determining if manufacturing system is performing well or not (Hopp and Spearman, 2011)

For the best case performance scenario it is assumed that a product does not have to wait before it can be processed. The time it takes for a product to get produced without waiting is defined as the raw processing time (T_0). The machine which takes longest to complete is the bottleneck of the manufacturing system. The rate it produces units is defined as the bottleneck rate (r_b). The WIP level (w) for which the bottleneck rate and raw processing time is given and the throughput is maximized is known as the critical WIP (W_0). This lead to the following equation

$$W_0 = r_b \cdot T_0$$

To determine the three scenarios Hopp and Spearman (2011) developed some formulas. For the best-case performance scenario the cycle time (CT_{best}) and throughput (TH_{best}) are:

$$CT_{best} = \begin{cases} T_0 & \text{if } w \leq W_0, \\ \frac{w}{r_b} & \text{otherwise.} \end{cases}$$

$$TH_{best} = \begin{cases} \frac{w}{T_0} & \text{if } w \leq W_0, \\ r_b & \text{otherwise.} \end{cases}$$

For the Worst-Case Performance scenario the cycle time (CT_{worst}) and throughput (TH_{worst}) are:

$$CT_{worst} = w \cdot T_0,$$

$$TH_{worst} = \frac{1}{T_0}.$$

And for the Practical Worst-Case Performance scenario the cycle time (CT_{PWC}) and throughput (TH_{PWC}) are:

$$CT_{PWC} = T_0 + \frac{w-1}{r_b},$$

$$TH_{PWC} = \frac{w}{W_0 + w - 1} \cdot r_b.$$

Using these formulas, the current performance of the production line can be determined, while taking the amount of operators into account. The throughput of every processing step can be calculated and with these results the number of operators can be determined.

Batching

Batching can be used to keep grids getting processed in the right order. Next to that batching can have a great influence on scheduling. When the batch sized is correctly chosen, it can keep cycle time low and due dates are more easily met (Hopp and Spearman, 2011). Hopp and Spearman (2011) defines the serial batch size as the number of jobs of a common family processed before the workstation is changed over to another family. In the case of Philips there are three types of families, e.g. Mammography-, Regular- and Round grids.

According to the law of process batching (Hopp and Spearman, 2011), it might be beneficial to have a batch size greater than one, cycle time grows proportionally with batch size and cycle time will be minimized for some batch size. To determined the optimal batch size Hopp and Spearman (2011) defined the following formulas.

$$t_e = s + k * t$$

$$u_0 = r_a * t$$

$$u^* = \frac{r_a}{k^*} \cdot t_e = \sqrt{u_0}$$

$$k^* = \frac{r_a \cdot s}{\sqrt{u_0} - u_0}$$

where, t_e is defined as the effective processing time. s is the setup time, k the size of the batch and k^* therefore the optimal batch size. t is the processing time per unit. r_a is defined as the arrival rate of units. u_0 is the utilization without setup and u^* the optimal utilization.

An easy way to reduce cycle time in factories is by reducing transfer batching. This is the number of parts that accumulate before being transferred to the next station (Hopp and Spearman, 2011). This is also something to keep in mind while considering batching as an option to prioritize orders within the production line. The average waiting time before a batch can go to the next step can be calculated using the following formula.

$$\text{Average wait to batch time} = \frac{k-1}{2 \cdot r_a}$$

3 Methodology

In this chapter, 4 different sections will be discussed. Section 3.1 explains the method to optimize stage 1. Section 3.2 shows a method how the DRC-FJSSP-SSO will be solved. The third section (Section 3.3), a method is presented to solve stage 3 of the production process and finally, a method is explained how the three stages can be combined. The interaction and deliverable of the four methods is visualized in Figure 12.

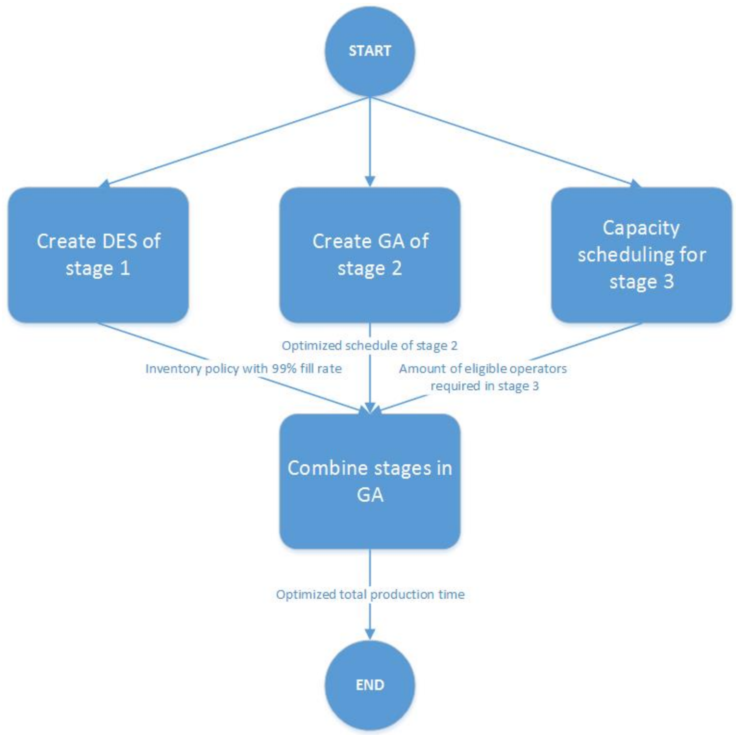


Figure 12: Interaction between methods

3.1 Stage 1 - Inventory Control

To manage the inventory of stage 1, a suitable inventory policy has to be determined. An inventory policy determines at which inventory level new products should be produced, how often the inventory should be reviewed and how many products should be produced at once. Section 2.1.1 discusses five different inventory policies. The policy which can be used by Philips is the (R, s, nQ) policy. This is the only policy that can be used due to two reasons: (1) Stage 1 produces in batches of nine fiber rolls at a time and (2) the 1D grids department does not have the right technology to keep continuous track of all inventory. The first reason forces the policy to produce in batches which means that nQ products can be produced per time. The lack of the right technology leads to the fact that inventory has to be checked manually. This can be done in different time periods. Therefore, it is important to know the review period R.

The (R, s, nQ) policy is now explained in more detail. First, the difference between IP and IOH has to be explained. The IOH is all inventory that is available within the location. This can be negative if products are demanded, but there is no more stock available. These are called backorders. IP is the IOH, but with all products which are in the system. This means that all products which are already in production are added to the IP. Figure 13 illustrates how the (R, s, nQ) policy works. Every R time units, the IP is reviewed. If the IP is below the reorder level (s), an order is placed to produce nQ products. The amount of products that will be produced depends on the number of products to exceed the reorder level once again and the batch size (Q). Let us consider the case where the reorder level is 22, the IP is 9, the IOH also has a value of 9 and the batch size is 5. If at this moment the IP is reviewed, it can be concluded that 13 products are required to meet the reorder level. However, the production occurs in batches of 5 products at a time. This means that 3 batches have to start in the production process. This will lead to the IP being updated to 24 and the IOH still being 9. After L time units the products are finished and the IOH is increased by 15 units. The review period and the reorder level are used to determine a suitable inventory policy for this production process.

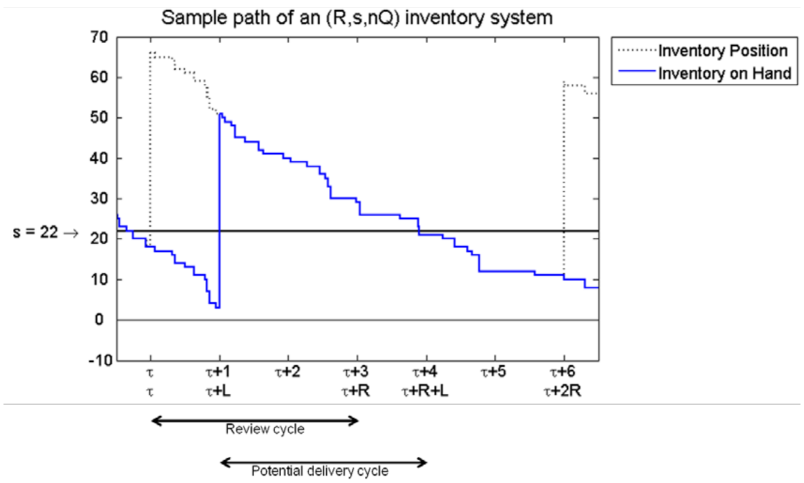


Figure 13: Example of an (R,s,nQ) inventory policy (van Donselaar and Broekmeulen, 2017)

To ensure that the optimal parameter (R and s) settings for the inventory policy are chosen, a Discrete Event Simulation (DES) is developed within the Python programming environment. The DES will test different parameter settings and find the right settings to meet the fill rate requirement of 99%. To create the DES, several steps are required. (1) the different steps of the production process have to be mapped. (2) Demand data has to be gathered and transformed to fit the simulation. (3) Processing times have to be determined. (4) The number of kached rolls from 1 container of fiber rolls has to be calculated. (5) The amount of grids produced from 1 kached roll has to be determined. (6) The production capacity for each process step has to be determined. Steps (2) - (5) might be challenging to perform due to huge variation in the production process, as well as the diversity of types of products.

The first step which is required for the creation of the DES has already been performed in Section 1.1.4. The DES will take five processing steps into account. In Appendix A, it is shown as stage 1. First, the fiber is dried. Next, the fiber is planed to the right thickness. The third step is applying shellac to one side of the fiber. Then the other side of the fiber is glued to a strip of lead. Finally, the glue has to dry for a couple of days.

The demand data, as mentioned in step (2), is gathered from an internal order list. In this list, all historic and future orders are displayed. There are several possibilities to use this data for the DES. The two most obvious methods are: the historic demand data can be used to forecast upcoming orders and the upcoming orders (including backlog) can be used to get an accurate view of the upcoming orders. Forecasting orders, however, is not suitable for this data set. This is because there is a lot of variation in the data set as visualized in Tables 2 and 3. Table 2 shows the coefficient of variance during the year is in all cases larger than 0.5, which makes it very hard to forecast. Next, Table 3 shows the increase and decrease of average demand per day over the years. It can be seen that almost every line type has a year where the demand is very large or low compared to its previous year. The demand data per line type is too variable. Therefore, forecasting is very hard, because it is unclear what the customers will be ordering next year. This is the reason why the demand, which is used in the DES, will be based on the historic data of previous months/years. By updating this data set every couple of months, the demand will be accurate most of the times. To get an indication how much grids are ordered per line type, Table 4 shows the average demand of last 2.5 year per line type. It can be seen that line type 50L is not ordered. For each of these line types, it is checked whether the demand follows a certain distribution. If so, this distribution is used within the DES. To fit certain distributions on the demand data, first, the parameters of the distribution have to be determined. Only distributions which have a boundary at 0 were taken into account. This is because the demand can not be less than 0. Next to that the distribution had to be discrete, due to the fact that demand can only be ordered in integers. For all line types, the Truncated Poisson distribution performed best. To see if the Truncated Poisson distribution fits the historic demand data, a Chi square test is performed. The results are shown in Table 5 and it can be concluded that none of the data of the line types fits with the Truncated Poisson distribution because the p-value is below 0.05. This is the reason why no distribution will be used in the DES to sample demand. Instead, historic demand data is used in the DES.

Table 2: Coefficient of variation of different years

	2019	2020	2021	2022
31L	1.53	1.00	1.43	0.70
36L	1.25	1.26	1.11	0.85
40L	0.66	0.62	0.57	0.54
41L	0.67	0.58	0.59	0.51
44L	0.61	0.62	0.52	0.44
50L	NA	5.23	9.06	4.36
52L	1.62	2.16	1.42	1.55
57L	9.00	6.35	4.06	3.00
60L	0.50	0.55	0.54	0.53
67L	0.81	0.69	0.59	0.62
70L	0.65	0.52	0.60	0.46
74L	0.98	0.74	1.04	0.77
80L	1.08	1.09	1.39	0.66
85L	2.08	2.78	4.18	1.72

Table 3: Increase/Decrease average demand per day per line type

	2019-2020	2020-2021	2021-2022
31L	4.7%	-10.9%	50.5%
36L	-6.5%	37.4%	-8.6%
40L	26.7%	34,9%	-49.4%
41L	-3.8%	-5.0%	5.6%
44L	-8.0%	47.1%	-28.5%
50L	NA	-75%	400.0%
52L	-39.7%	201.6%	-22.1%
57L	87.5%	246.7%	13.5%
60L	5.2%	-7.7%	31.4%
67L	8.8%	12.1%	-24.9%
70L	13.6%	-36.4%	115.3%
74L	65.2%	-42.4%	130.0%
80L	32.1%	37.8%	220.0%
85L	-24.5%	-32.4%	332.0%

Table 4: Grids required per day per line type

31L	**36L**	**40L**	**41L**	**44L**	**50L**	**52L**	**57L**	**60L**	**67L**	**70L**	**74L**	**80L**	**85L**
6	3	15	16	16	0	2	1	20	3	21	6	4	2

Table 5: Chi squared - test

	31L	**36L**	**40L**	**41L**	**44L**	**50L**	**52L**	**57L**	**60L**	**67L**	**70L**	**74L**	**80L**	**85L**
statistic	380.8	265.3	2046.4	1609.9	2999.9	120.1	134.6	168.2	2736.9	1382.8	1392.2	1418.5	325.4	186.2
p-value	3.1e-20	6.3e-10	1.9e-279	1.5e-204	0.0	3.2e-4	3.6e-4	2.8e-5	0.0	1.47e-165	8.2e-162	4.2e-174	1.4e-12	7.4e-5

The processing times of each step are not registered in a database. However, there are some timestamps registered on paper and in a database. This is the case for the starting time of the planing process and the end moment of applying the lead. These time points can give to a certain extent an indication of what the processing times are. Next to this data, there are also very experienced operators. The processing time indications are therefore checked by the experienced operators to see if they are accurate. The time of the drying steps is pre-determined. The first drying step takes 3 days if the container is placed in an oven. If not, the container has to stay in the factory for more days, but the length of this stay depends on the line type. The processing/waiting times that are used are presented in Table 6. The Processing time is the time from planing the fiber until the lead is glued on the fiber including possible waiting times between the processes.

Table 6: Average lead times per line type in days

	31L	**36L**	**40L**	**41L**	**44L**	**50L**	**52L**	**57L**	**60L**	**67L**	**70L**	**74L**	**80L**	**85L**
Process time	3.8	3	5	5.7	4.8	4.8	4	8.7	12	9.3	8.7	9.2	6.3	9.2
Wait time (pre)	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Wait time (after)	3	3	3	3	13	3	3	3	3	3	3	3	3	3

Since the rolls of fiber are not always similar in size, it is challenging to predict how many grids a container of fiber will produce. The team leads have a document containing a rough indication of these numbers. Together with several experienced operators and engineers, this document has been checked and this led to the values presented in Table 7.

Table 7: Average number of kached rolls per container of fiber

31L	**36L**	**40L**	**41L**	**44L**	**50L**	**52L**	**57L**	**60L**	**67L**	**70L**	**74L**	**80L**	**85L**
9	9	9	9	14	14	14	10	20	20	20	20	10	10

The fifth data requirement is the number of grids that can be produced with one kached roll. Again there is much variation in this step. The grids can for example differ in size. Together with a team lead, operators and engineers, the following data was gathered. On average one kached roll will result in 3 grids.

Table 8: Average number of grids per kached roll

31L	**36L**	**40L**	**41L**	**44L**	**50L**	**52L**	**57L**	**60L**	**67L**	**70L**	**74L**	**80L**	**85L**
3	3	3	3	4	3	3	3	3	4	4	3	3	3

At last, the capacity per production step has to be determined. The planing process contains 2 machines that work in parallel. Besides, the morning and midday shifts always devote an operator to this process. This means that 80 hours per week are used for planing. In these 80 hours, the 2 machines can produce 10 containers with different types of fiber. This is directly the process step that has the smallest production capacity. In the next two steps, 2 containers can be processed per shift (consists of 8 hours). This results in a capacity of 20 containers per week if 2 shifts are used per day.

Finally, the fill rate has to be calculated to see if the preferred 99% is reached. This can be done by dividing the time where the IOH is positive by the total time. If the total time (simulation length) is very large, the fill rate will converge to its steady state. This is useful, because this makes the fill rate an important result. By changing the review period and the reorder level, the fill rate will be influenced. For example, if the reorder level is 5, the review period is 1 month and the demand per week is 10 items, a low fill rate will be the result. Because the demand in this example is very large compared to the low reorder level and long review period, it will lead to a very low fill rate. Increasing the reorder level or shortening the review period can increase the fill rate. The optimal review periods and reorder levels per line type can be determined by systematically checking different combinations. Besides that, also a varying demand is compared to see if this impacts the two parameters.

3.2 Stage 2 - DRC-FJSSP-SSO

In the next stage, a Dual Resource Constraint Flexible Job Shop Scheduling Problem with Scarce Setup Operators (SSO) model is developed, based on (Obimuyiwa, 2020). A MILP is used to describe the model. Two methods are used to solve this model. First, the MILP is solved using an LP solver and afterwards, a GA is used to solve the model. Two different methods are used because the LP solver is expected to be unable to solve the problem within a reasonable computation time. The method for developing and solving a MILP is explained in Section 3.2.1 and the method of how a GA is used to solve this problem is described in Section 3.2.2.

The functioning of the DRC-FJSSP-SSO is visualized in Figure 14. It works as follows, each Job consists of several Operations. In the case of Philips, each Job consists of 3 Operations: Matten, Plakken and MLM respectively Operation 1, Operation 2 and Operation 3. These Operations have to be processed in this sequence. Every operation has to be processed on a machine. It can be the case that there is only 1 machine applicable for an operation, but it can also be that there are multiple eligible machines. For each operation performed on a machine, an eligible operator will perform a setup on the machine. When this setup is done, the operation can start and the operator does not have to monitor the machine anymore. If an operation is completed, the next operation can start if an eligible machine and eligible operator are available. Every machine performs runs. This means, when a machine is processing its first job, this is its first run. When this job is finished, the second run can start for this machine. The same goes for the operators. But for operators, the runs are setups. This means an operator can perform setup multiple setups consecutively.

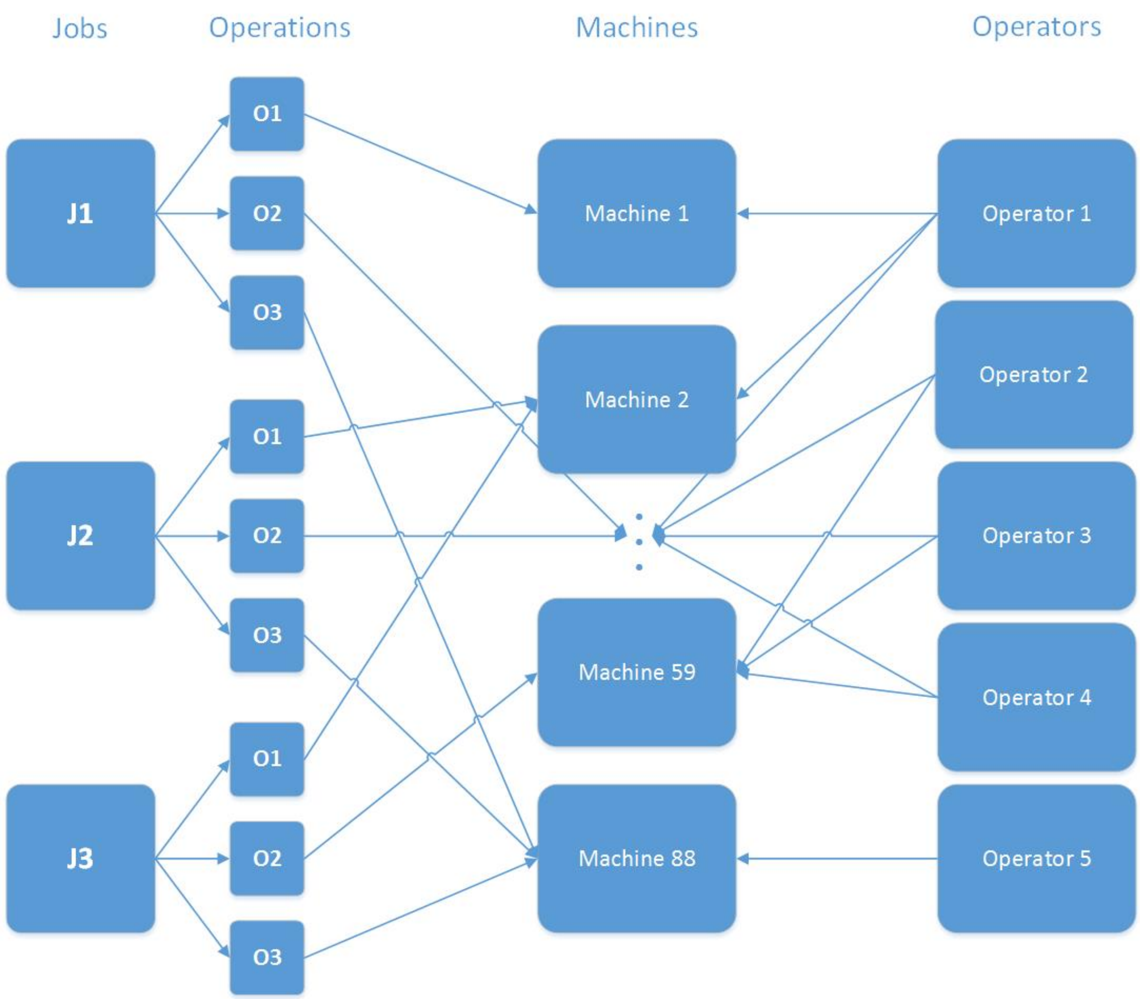


Figure 14: DRC-FJSSP-SSO visualized

The parameters, processing time, setup time, eligible machine, eligible operators, and demand have to be determined beforehand and can be found in Appendix C. The values for the processing time of the first and third operations are based on operator experience. The processing times for these operations are grid-independent. For the second Operation, the processing time is grid-dependent. Each machine has its own time in which it glues 1 lamellae to the previous one. This is in all cases between 4 and 8 seconds. This time is manually determined using a stopwatch because no better data was available. The time of glueing 1 lamellae is then multiplied by the number of lamellae, which differs per type of grid.

Furthermore, the setup time is also not known beforehand. That is why these times also are based on operator experience. First the different setups have to be determined. For Operation 1, there are 3 different setup options. First, the grid type which is being produced is the same as the previous grid which is produced on that machine. Second, the line type differs from the previous and current grid types. At last, the height of the previous and current grid types are different. For the second operation, there are 2 different types of machines, one that uses the STB method and one that uses the LTB method. The machines which produce using the STB method have the same setup options as the machines of Operation 1. This means the grid type is the same, the line type differs, or the height differs. The machine which produces using the LTB method only has 1 setup time. The same goes for the machine for Operation 3. The time belonging to these different types of setups are displayed in Appendix C.

The list of eligible machines for each operation and eligible operators for each machine can also be found in Appendix C. The demand data which is used is again based on backlog and real future demand for the upcoming 6 weeks.The real future demand is known for 6 weeks, because products are ordered beforehand.

3.2.1 MILP

The mathematical model of Obimuyiwa (2020) is used to replicate the production process of Philips. Several changes have been made to the model. First, 2 mistakes were resolved in the model, and second, 2 extra constraints have been added to create a variable setup time. The complete model is shown and explained below.

Sets

R_m: Set of number of production runs per machine $(1, \ldots, R_m)$,
M: Set of Machines
O: Set of Operations
J: Set of Jobs $(1, \ldots, J)$,
K_n: Set of number of setups per Operator $(1, \ldots, K_n)$,
N: Set of Operators

Parameters

R_m: The maximum number of runs for machine m,
K_n: The maximum number of setups for Operator n,
$P_{o,j,m}$: Is 1 if operation o of job j can be processed on machine m, 0 otherwise,
$\Theta_{n,m}$: Is 1 if operator n is eligible to operate machine m , 0 otherwise,
Ω: Large positive number (at least twice as large as the job with the longest processing time for Operation 2),
$H_{m,j,j'}$: Setup time of machine m from job j to job j' ,
$T_{o,j,m}$: Processing time of operation o of job j on machine m

Decision Variables

$\hat{C}_{r,m} \geq 0$: Completion time of the r^{th} run on machine m,

$C_{o,j} \geq 0$: Completion time of operation o of job j,

$s_{k,n} \geq 0$: Completion time of k^{th} setup of operator n,

$C_{max} \geq 0$: Makespan, i.e. completion time of all jobs,

$S_{r,m,o,j} \geq 0$: Setup time for operation o of job j for the r^{th} run on machine m,

$$x_{r,m,o,j} = \begin{cases} 1, & \text{if operation } o \text{ of job } j \text{ for the } r^{th} \text{ run on machine } m \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{k,n,r,m} = \begin{cases} 1, & \text{if the } k^{th} \text{ setup of operator } n \text{ is for the } r^{th} \text{ run on machine } m \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

$$z_{r,m} = \begin{cases} 1, & \text{if the } r^{th} \text{ run of machine } m \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

$$g_{k,n} = \begin{cases} 1, & \text{if the } k^{th} \text{ setup of operator } n \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

Next, the model itself is explained per constraint. Equation (5) is the objective function of this model. It states to minimize the makespan.

$$Min \quad C_{max} \tag{5}$$

Subsequently, equation (6) makes sure that the makespan is the time at which the last operation is finished.

$$C_{max} \geq C_{o,j}, \qquad \forall (o,j), \tag{6}$$

In constraints (7) and (8) it stated that if operation o of job j is running on machine m on the r^{th} run, the time of machine m when it performs the r^{th} run is equal to the time at which operation o of job j is completed.

$$\hat{C}_{r,m} \geq C_{o,j} + \Omega \cdot x_{r,m,o,j} - \Omega, \qquad \forall (r,m,o,j), \tag{7}$$

$$\hat{C}_{r,m} \leq C_{o,j} - \Omega \cdot x_{r,m,o,j} + \Omega, \qquad \forall (r,m,o,j), \tag{8}$$

Constraint (9) shows that if operation o of job j is running on machine m on the 1^{st} run, the completion time is equal to the Processing time combined with the setup time.

$$\hat{C}_{1,m} - T_{o,j,m} - S_{1,m,o,j} + \Omega \cdot (1 - x_{1,m,o,j}) \geq 0, \qquad \forall (m,o,j), \tag{9}$$

Constraint (10) is similar, but it explains what happens to the completion time if it is not the first run. In that case, the completion time is equal to the completion time of the previous run on that machine plus the processing time and setup time.

$$\hat{C}_{r,m} - T_{o,j,m} - S_{r,m,o,j} + \Omega \cdot (1 - x_{r,m,o,j}) \geq \hat{C}_{r-1,m}, \qquad \forall (r,m,o,j) | (r > 1), \tag{10}$$

In Equation (11) a mistake within the model from (Obimuyiwa, 2020) was corrected. The plus sign was a minus, but this led to decision variables becoming as large as Ω. By changing it to a plus sign, the constraint works as it should. This is, a consecutive Operation of a Job can start at the time point where the first operation is finished.

$$\hat{C}_{r,m} - T_{o,j,m} - S_{r,m,o,j} + \Omega \cdot (2 - x_{r,m,o,j} - x_{r',m',o-1,j}) \geq \hat{C}_{r'-1,m'}, \tag{11}$$

$$\forall (r,m,r',m',o,j) | ((r,m) \neq (r',m')) \wedge (o > 1),$$

Equation (12) shows that the completion time of the k^{th} setup of operator n is equal to the r^{th} run of machine m minus the processing time, if the k^{th} setup of operator n is used for run r on machine m and if operation o of job j is running on machine m on run r^{th} run.

$$s_{k,n} - \Omega \cdot (2 - x_{r,m,o,j} - y_{k,n,r,m}) \leq \hat{C}_{r,m} - T_{o,j,m}, \quad \forall(r, m, o, j, k, n), \tag{12}$$

The next two constraints make sure that the completion time of the k^{th} setup of operator n is larger than the setup time (if it is the first run of the machine) or the setup time plus the completion time of the previous run of that machine.

$$s_{k,n} - S_{1,m,o,j} + \Omega \cdot (2 - x_{1,m,o,j} - y_{k,n,1,m}) \geq 0, \quad \forall(m, o, j, k, n), \tag{13}$$

$$s_{k,n} - S_{r,m,o,j} + \Omega \cdot (2 - x_{r,m,o,j} - y_{k,n,r,m}) \geq \hat{C}_{r-1,m}, \quad \forall(r, m, o, j, k, n)|(r > 1) \tag{14}$$

Equations (15) and (16), explain that the completion time of the k^{th} setup of operator n is larger than the setup time (if it is the first setup of the operator) or the setup time plus the completion time of the previous setup of the operator.

$$s_{1,n} - S_{r,m,o,j} + \Omega \cdot (2 - x_{r,m,o,j} - y_{1,n,r,m}) \geq 0, \quad \forall(r, m, o, j, n), \tag{15}$$

$$s_{k,n} - S_{r,m,o,j} + \Omega \cdot (2 - x_{r,m,o,j} - y_{k,n,r,m}) \geq s_{k-1,n}, \quad \forall(r, m, o, j, k, n)|(k > 1), \tag{16}$$

Equation (17) verifies that when operation o of job j is running on machine m on the r^{th} run, it can only be active if operation o of job j can be processed on machine m.

$$x_{r,m,o,j} \leq P_{o,j,m}, \quad \forall(r, m, o, j), \tag{17}$$

The next constraint is quite similar, since it makes sure that when setup k of operator n is processed on machine m on the r^{th} run, it can only be active if operator n has the skills to use machine m.

$$y_{k,n,r,m} \leq \Theta_{n,m}, \quad \forall(k, n, r, m), \tag{18}$$

Equation (19) ensures that operation o of job j can only be processed on 1 run on a machine.

$$\sum_{m=1}^{M} \sum_{r=1}^{R_m} x_{r,m,o,j} = 1, \quad \forall(o, j), \tag{19}$$

The next constraint explains which run of a machine is selected. Equation (21) continuous on constraint (20) and states that a run on a machine can only be used if the previous run is also used.

$$\sum_{j=1}^{J} \sum_{o=1}^{O_j} x_{r,m,o,j} = z_{r,m}, \quad \forall(r, m), \tag{20}$$

$$z_{r,m} \leq z_{r-1,m}, \quad \forall(r, m)|(r > 1), \tag{21}$$

For the next constraint, Equation (22), again a mistake was removed from the model of (Obimuyiwa, 2020). The variable after the equal sign first stated $g_{k,n}$. This is corrected to $z_{r,m}$. This is, because the sum of all possible setups which can be performed by an operator on a certain machine and run, can only be 1 if run r on machine m is used.

$$\sum_{n \in N} \sum_{k \in K_n} y_{k,n,r,m} = z_{r,m}, \quad \forall(r, m), \tag{22}$$

Equation (23) makes sure that setup k of operator n can at most be used once for all runs on all machines.

$$\sum_{m=1}^{M} \sum_{r=1}^{R_m} y_{k,n,r,m} = g_{k,n}, \quad \forall(k, n), \tag{23}$$

The next constraint shows that a setup of a certain operator can only be done if (s)he already did their previous setup.

$$g_{k,n} \leq g_{k-1,n}, \quad \forall (k,n) | (k > 1), \tag{24}$$

Equations (25) and (26), guarantee that the Operation sequence has to be kept for all x variables. This means that if a previous Operation is performed, an upcoming Operation cannot be performed on that same machine in an earlier run.

$$x_{r',m,o',j} \leq 1 - x_{r,m,o,j}, \quad \forall (r, r', m, o, o', j) | (o' < o) \wedge (r' > r), \tag{25}$$

$$x_{r',m,o',j} \leq 1 - x_{r,m,o,j}, \quad \forall (r, r', m, o, o', j) | (o' > o) \wedge (r' < r), \tag{26}$$

The same is done for the y variable by including Equation (27) and (28). Here, if a previous setup is performed, an upcoming setup cannot be performed on that same machine in an earlier run and vice versa.

$$y_{k',n,r',m} \leq 1 - y_{k,n,r,m}, \quad \forall (k, k', n, r, r', m) | (k' < k) \wedge (r' > r), \tag{27}$$

$$y_{k',n,r',m} \leq 1 - y_{k,n,r,m}, \quad \forall (k, k', n, r, r', m) | (k' > k) \wedge (r' < r), \tag{28}$$

Equations (30) and (29) are newly added constraints. Equation (29) makes sure that in the first run, the setup time is always equal to 5 since no starting configuration is taken into account. This is based on the fact that the operators has to start the machine. Equation (30) ensures that the setup time of the new job depends on the job that was processed in the previous run on that specific machine. Therefore, this developed model can take machine configuration into account. The setup times on a machine are not always the same but can differ based on previous and current type of produced product. When reviewing literature, the combination of having a DRC-FJSSP-SSO with a setup time dependant on the previous production run is not found. This is due to the fact that a DRC-FJSSP-SSO is not common in literature. Therefore, this development cannot be found.

$$S_{1,m,o,j} = 5, \quad \forall (m, o, j), \tag{29}$$

$$H_{m,j,j'} \cdot (x_{r,m,o,j} + x_{r+1,m,o,j'}) - H_{m,j,j'} \leq S_{r+1,m,o,j'}, \quad \forall (r, m, o, j, j') | (r > 1) \wedge (j \neq j'), \tag{30}$$

Finally, Equation (31) tells that x, y, z and g are binary variables.

$$x_{r,m,o,j}, y_{k,n,r,m}, z_{r,m}, g_{k,n} \in \{0, 1\}, \quad \forall (r, m, o, j, k, n) \tag{31}$$

To summarize the mathematical model,

$$\begin{aligned} Min \ \ & C_{max}, \\ s.t. \ \ & (6) - (31). \end{aligned}$$

The MILP will be solved using PulP in Python. PulP is an Linear Programming (LP) modeler and is used to program this mathematical model. Within PulP several solvers can be used like, GUROBI, CPLEX and GLPK. The computation time when using a different solver can differ quite much, so multiple solvers will be tested. The MILP is solved using PulP to find an optimal solution for this problem. In Section 4.2.1, it can be seen that solvers require a lot of computation time to find optimal results. In order to investigate whether the optimal solution can be found with a method requiring less computational time, a GA is modeled in the next phase.

3.2.2 GA

GAs were introduced by Holland et al. (1975) and are based on concepts of evolution and natural selection. The idea is to create random solutions for a given optimization problem and "evolve" the solutions towards an optimal solution. The optimal solution is in this case based on the selection pressure induced by the objective function. A GA can be very effective when handling large search spaces (Nobile, 2021). It is are therefore not used to find an optimal solution, but rather to find "good" solutions for a very complex problem. How GAs exactly work is explained in the steps below and visualized in Figure 15. Next, every step within the iterative cycle will be explained in more detail. These steps are based on Nobile (2021).

1. (Population) N "random" individuals are created: the population
2. (Fitness evaluation) The fitness value for all individuals is determined
3. (Termination criterion) If the termination criterion is met, the algorithm will stop and the solution will be the best fitting individual. Else, go to step 4
4. (Selection of parents) A selection mechanism is used to create pairs of individuals (with a probability proportional to their fitness value)
5. (Crossover) The selected pairs exchange "chromosomes" to form new individuals
6. (Mutation) The new individuals mutate
7. (New Offspring) The new individuals, i.e. offspring, replace the old population, go to step 2

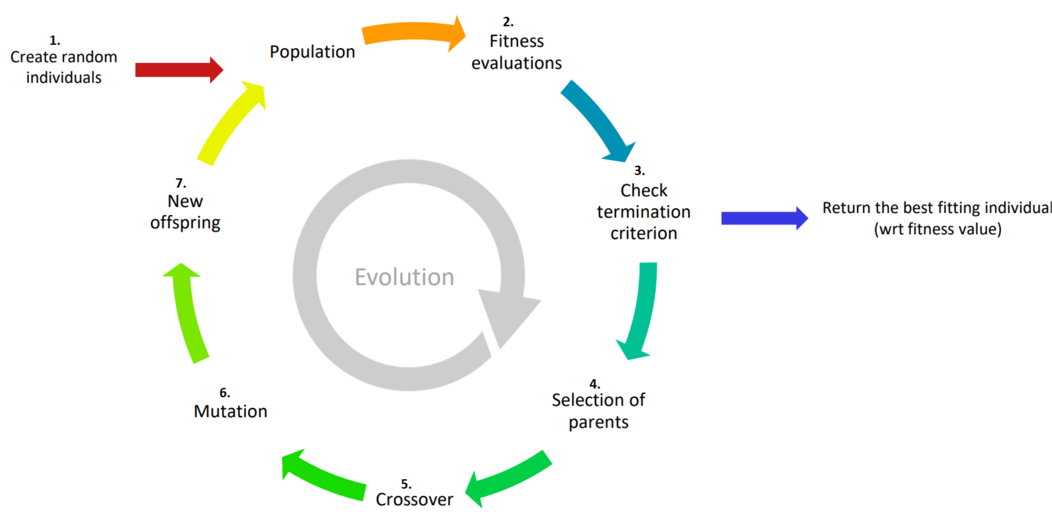


Figure 15: Genetic Algorithms iterative process (Nobile, 2021)

1. Create initial population
In most cases, the population of N individuals is created randomly. This is done using a uniform distribution or via some information that is already known. An advantage of using the uniform distribution is that no bias is introduced in the process. A disadvantage is that it might take longer to converge to the optimal solution. A biased starting population might converge sooner, but it might be the case that the population becomes too biased and all variation is gone. For a DRC-FJSSP an individual contains multiple genes. A single gene looks like a list of 4 numbers, (j, o, m, n), where j is formulated as the job, o as the operation of the job, m is the machine where the process takes place and n is the operator which is required to process the task. This means that if there are 3

jobs and each job contains 4 operations, an individual contains 12 genes.

Bias can be added to the starting population by changing the order of the genes. In this thesis several biased and a random starting populations are tested. First, genes are tested where they represent a sequence of operations, see Figure 16. This means that all genes for Operation 1 are first in the schedule, next up the genes for Operation 2 will follow, and finally the genes for Operation 3. This bias is chosen, because first, all prior operations are scheduled before scheduling the next operation which will decrease the waiting time of operations. This means that first, Operation 1 from every job will be scheduled. Next, Operation 2 of every job and finally, Operation 3 of all jobs.

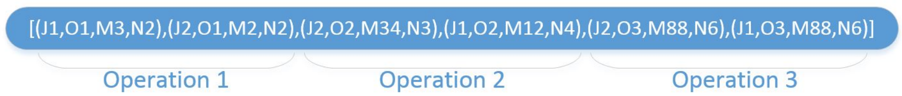


Figure 16: Biased starting population: Genes are in sequence of operation

The second biased starting population that is used is again arranging the genes based on operation, but also on the job, see Figure 17. This means that first job x will be scheduled all the way from Operation 1 until Operation 3. Next, all operations for another job are scheduled. This is again done to reduce waiting times. If an operation is scheduled right after another operation, it does not have to wait (if there are machines and operators available) to be processed.

Figure 17: Biased starting population: Genes are in sequence of job and operation

The random schedules have only one constraint, which is that operations have to be arranged in the right order. This means, first Operation 1 has to be scheduled, which is followed up by Operation 2 and finally Operation 3. A possible schedule can thus look like Figures 16 and 17, but different configurations, like Figure 18, are also possible.

Figure 18: Random starting population

A combination of these different types of individuals will also be used as a starting population.

2. Fitness evaluations

In the population, every individual has to calculate the fitness value. In the case of a scheduling problem, this might be the makespan or the number of tardy jobs. In this thesis, it is chosen to minimize the completion time of every operation of every job. Although the makespan of a solution is the final goal, this is not minimized. This is since minimizing the makespan takes a lot of computation time and the GA will be more likely to be stuck in a local optimum. The reason why minimizing the total completion time of each operation does not often get stuck in the local optimum, is the fact that there are a lot of possibilities to reduce this objective. If one operation is changed in the schedule, the objective will probably differ. It is, therefore, easier to reduce this objective. The makespan cannot easily be minimized, due to the fact that only the completion time of the last operation has to be decreased to minimize the objective. It is therefore much harder to reduce this objective in a large and complex problem. Maassen et al. (2020) showed that minimizing the total completion time, performed very well on minimizing the core waiting time. Minimizing the core waiting time led moreover to a lower makespan. This means that minimizing the completion time, will in most cases lead to a lower makespan.

3. Termination criterion

In most real-world scenarios, the optimal fitness value is unknown. This means that the final solution might not be the optimal solution. To still get a valid solution three termination criteria are mentioned: (1) terminate as soon as the fitness value reaches a user-defined threshold, (2) terminate after a fixed amount of generations or (3) terminate when the population loses diversity.

4. Selection of parents

Parent selection implements the "survival of the fittest". This selection process must be done carefully. Removing all "bad" solutions too fast can lead to a loss of diversity. This can result in a final solution being at a local optimum. The methods that are used to select parents are problem-dependent. For this problem, tournament selection is used. It works as follows. k unique individuals are chosen from the population to participate in the tournament. The individual with the best fitness value within the tournament wins the tournament and is therefore selected. An advantage of this method is that every individual can be chosen. Moreover, the selection pressure can be controlled using parameter k. A large k will lead to "bad" individuals having a smaller possibility of being chosen.

5. Crossover

Crossover means that the chosen parents mate and create offspring/individuals by exchanging a part of their genes. The underlying idea is those sub-optimal parents contain good patterns within their genes. These patterns, once combined with the other parent might lead to "excellent" new individuals. The crossover occurs with a crossover probability (P_c). If parents do not undergo crossover they are the offspring themselves. The main objective of this crossover is not just to produce new individuals with a better fitness, but also to get individuals which are valid based on the problem type (Sivanandam and Deepa, 2008; Talbi, 2009)). Kacem et al. (2002), Lee et al. (1998) and Defersha and Chen (2010) used crossovers which can be used for a DRC FJSSP. The crossover operators are called: Machine-operation crossover operator(MO), Job-operation sequence-order crossover operator(JOSO), and single-point crossover operator(SP). For a population, without any bias, these are good operators, but if bias is introduced in the starting population it is also required to keep this bias. Otherwise, the schedule will become infeasible. That is why different crossover operators are used when using a biased starting population. The crossover operator that will be used in this thesis is the JOSO crossover operator. This is because the sequence of the genes is very important to increase the fitness and the JOSO operator will keep this into account. The steps performed by the JOSO crossover operator are explained below and visualized in Figure 19. It can be seen that offspring one gets random genes from its first parent, next this offspring receives its genes from the second parent in such a way that the schedule is feasible.

1. Two new individuals are created using random genes of one of their parents (see left side of Figure 19)
2. The rest of the genes are retrieved from the other parent in the same sequence as its parent.
3. The machine-assignment value is retrieved from the other parent and a new machine-assignment value is assigned (see right side of Figure 19)

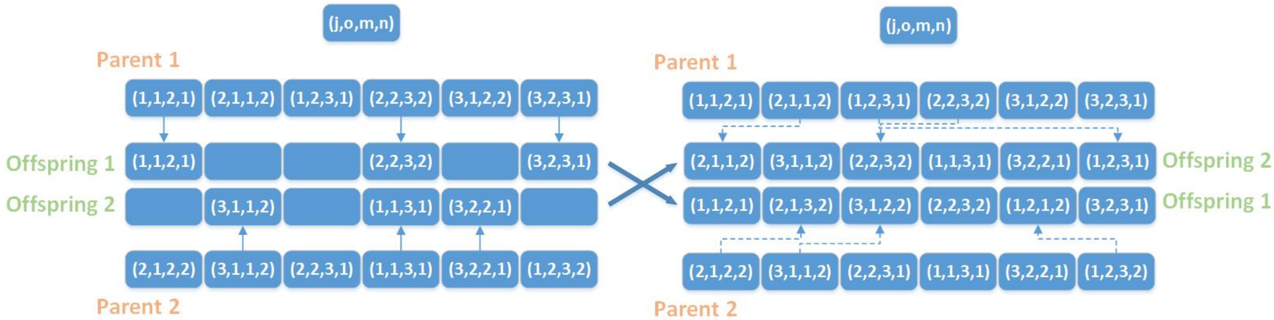


Figure 19: Job-operation sequence-order crossover

For the biased schedules, the crossover works similarly. Figure 20 shows when genes are in sequence of operation. In this case, the genes of operation 1 can only be in the first half of the schedule, and the genes of operation 2 can only be in the second half of the schedule. This will lead to the fact that all operations 1 are first fully scheduled and then operation 2 can start. The other biased schedule, where genes are in sequence of job and operation, will switch complete jobs, see Figure 21. That means that Operations 1, 2 and 3 are all replaced if that has to happen.

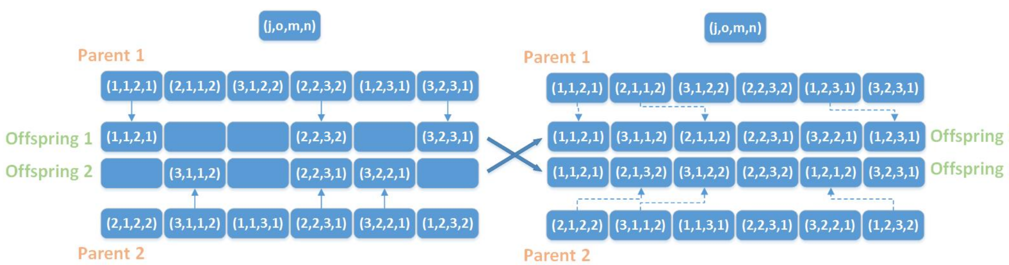


Figure 20: Crossover operator for genes in a sequence of operation

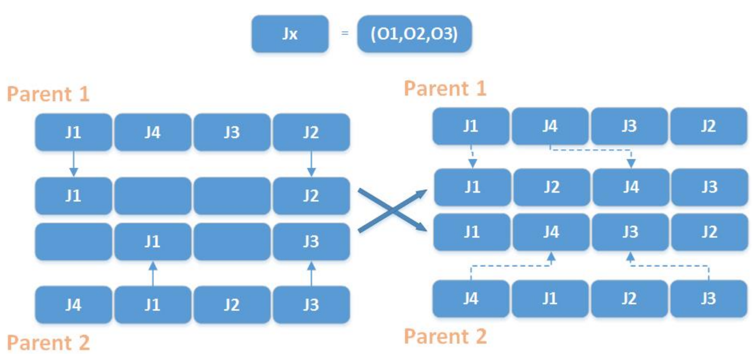


Figure 21: Crossover operator for genes in a sequence of job and operation

6. Mutation
A mutation is used to introduce new genetic material into the population. By doing this a population can get out of a local optimum. The mutation can lead to big jumps within the search space. The probability of a mutation is set to be P_m. Multiple mutations can be used but the most frequently used one is the uniform mutation. It picks a random gene and assigns a new random variable. In the case of the genes that are used in this thesis, (j, o, m, n), the machine and operator

are changed and two genes switch places.

7. New offspring
After the mutation might have taken place, the new offspring is complete and a new population is formed. This means the iterative cycle starts again until the termination criterion is met.

Optimizing Genetic Algorithm
Trying to optimize the GA is hard. Hyper-parameters are therefore of great importance. By choosing the wrong values, the population might converge too fast or diversity can be lost. A poorly chosen mutation probability can for example lead to not being able to escape local optima, or not being able to preserve a good scheme.Therefore, the hyper-parameters have to be tested for different values to determine which values lead to the best result. The value for each hyper-parameter is problem-dependent (Nobile, 2021). This means it cannot be stated beforehand that a certain value will work. The hyper-parameters that will be tested are the tournament size, the crossover probability and the mutation probability.

Modeling
The GA is programmed in Python using the DEAP module (Fortin et al., 2012). This module contains the basics of a GA, but you can use your own operators. This is useful because different crossover- and mutation operators are used.

3.3 Stage 3 - Capacity Planning

For the final stage, the maximum capacity of each process is determined based on the number of available operators. First, all activities that have to be done by operators per processing step have to be captured. Second, activities have to be checked for constraints. Herewith, certain activities require multiple machines or operators. Next, for all activities, the time it takes has to be mapped. This can be achieved by checking how long it takes to process one grid, or how long this activity takes per shift. When all activities have corresponding times, the happy flow of this stage can be determined. At Philips, there are three different main grid types: Regular-, Mammography- and Round grids. Each of them has a different routing through this stage. That is why they will have different production times.

For each process that might be a bottleneck (Post-processing 1, Veneer and Post-processing 2), different configurations of operators will be examined. For example, configuration 1-2-1 means that there is one operator during shift 1, two operators during shift 2 and again one operator during shift 3. Now it can be calculated how much time per shift is available to perform all activities. By computing the available time per shift, it can be determined how many grids can be processed within 24 hours in this stage. Besides, with this knowledge, the bottleneck of these three processes can be determined.

Using the bottleneck rate, the performance of stage 3 can be examined based on throughput and cycle time. Hopp and Spearman (2011) developed simple formulas, based on Little's Law (Little, 1961) to examine the performance. In total, Hopp and Spearman (2011) defined three scenarios: best case, worst case and practical worst case. The time it takes for a product to get produced without waiting is defined as the raw processing time (T_0). The machine which takes the longest to complete is the bottleneck of the manufacturing system. The minimum rate at which a machine produces products is defined as the bottleneck rate (r_b). The WIP level (w), for which the throughput is maximized is known as the critical WIP (W_0). This leads to Equation (32):

$$W_0 = r_b \cdot T_0 \tag{32}$$

For the best-case performance scenario, it is assumed that a product does not have to wait before it can be processed. The formulas for the best-case performance scenario concerning the cycle time

(CT_{best}) and throughput (TH_{best}) are:

$$CT_{best} = \begin{cases} T_0 & \text{if } w \leq W_0, \\ \frac{w}{r_b} & \text{otherwise.} \end{cases}$$

$$TH_{best} = \begin{cases} \frac{w}{T_0} & \text{if } w \leq W_0, \\ r_b & \text{otherwise.} \end{cases}$$

For the Worst-Case Performance scenario, it is assumed that every order waits for its previous order to be finished before it can start. The formulas to determine the cycle time (CT_{worst}) and throughput (TH_{worst}) are:

$$CT_{worst} = w \cdot T_0,$$
$$TH_{worst} = \frac{1}{T_0}.$$

And for the Practical Worst-Case Performance scenario is based on the average waiting time at each processing step. The formulas for the cycle time (CT_{PWC}) and throughput (TH_{PWC}) for this scneario are:

$$CT_{PWC} = T_0 + \frac{w-1}{r_b},$$
$$TH_{PWC} = \frac{w}{W_0 + w - 1} \cdot r_b.$$

If the actual throughput or cycle time is between the best-case scenario and the practical-worst-case scenario, the production line is lean. For these formulas to hold, the production line has to be stationary. This is however not the case currently. Philips is constantly trying to improve the throughput and decrease the cycle time of the line. This is why these formulas cannot be used.

3.4 Combining Stages

The last part of this thesis will combine the three stages. First stage 1 is considered. If Philips will use the inventory policy that will be determined, the fill rate would be at least 99%. Knowing this, it can be assumed that there is always inventory available to start the production process in stage 2.

Stages 2 and 3 will on the other hand be combined within one model. The GA from stage 2 can be used for this. Implementing stage 3 into the GA is done by assuming that stage 3 is one unit as can be seen in Appendix B. The number of operations is increased by one. This operation, Operation 4, is called Post-processing. Operation 4 will differ in production time based on the characteristics of the grid. This is since regular-, mammography- and round grids have a different routing in stage 3. The number of operators, which are required for this stage, can be determined using the capacity scheduling. Different scenarios can be used, which means that a different number of operators are used.

Subsequently, yield is taken into account during Post-processing. Quality check 1 has an average yield of 78.8% and Quality check 2 has an average yield of 77%. The percentages can be used to determine how many grids will on average be in Operation 4 and how many grids on average will be completed. Based on the number of grids that have to be processed in Operation 4, it can be determined how many operators are required within Operation 4.

At last, the average total production time of a grid can be determined. This can be done using the results of the adapted GA and the results of stage 3. A Markov chain can be created as can be seen in Figure 22. Using the Markov chain, the steady-state of the total production time can be determined while taking into account the yield (Prob) and production times per processing step

(Pt). The current average production time of the complete production line is then compared to the average production time when entering the newly found processing times.

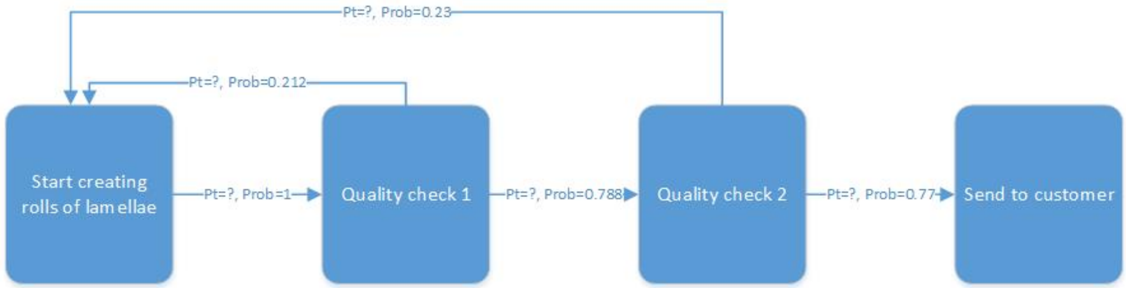


Figure 22: Markov chain of stage 2 and stage 3

4 Results

In this section the results will be presented for all three stages separately and combined. First the results of the DES will be explained. Next, the results of the MILP and GA will be discussed. As third, the results of stage 3, capacity scheduling, will be reviewed. Finally the models are combined and evaluated.

4.1 Inventory Control

A simulation should have a warm-up period. In this period, the simulation does not show the correct results, because it is not fully active yet. Figure 23 illustrates this. Right at the beginning of the simulation the IOH drops. This is because in the beginning nothing is being produced. Once the first batch of products is being produced, it still takes some time (waiting time + processing time) until it is ready to use. During this period demand keeps coming which led to the IOH being far below zero. To fix this problem, inventory/WIP was added at the beginning of a simulation. In that case, see Figure 24, the IOH will not drop, because products are still being produced. It can be seen that the fill rate is not 100%, because the IOH will still drop in some cases. For this example, a simulation length of 1 year is chosen. The green line in Figure 24 represents the reorder level and if the IOH goes below the red line there is not enough inventory. The policy $(1, 30, n*27)$ represents the (R, s, nQ) policy. R is the review period of 1 day, s is the reorder level of 30 and Q is 27, which is the amount that has to be produced per batch.

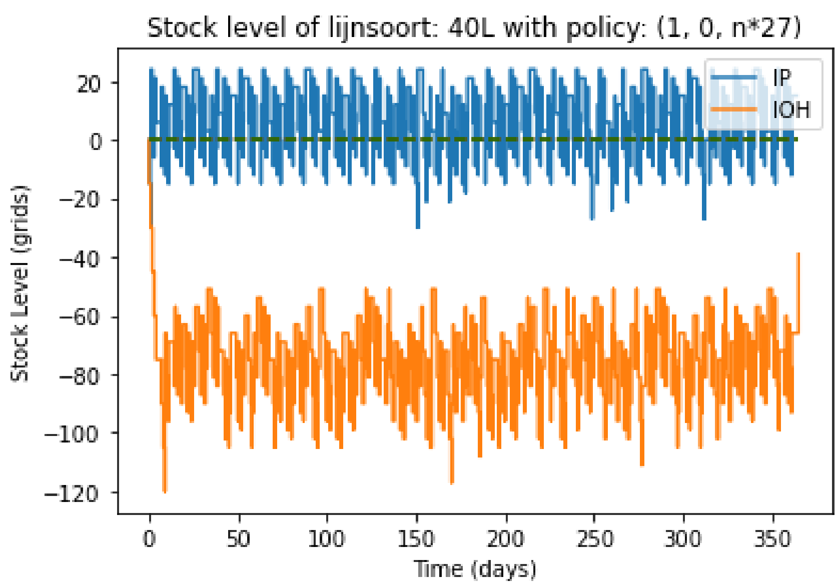


Figure 23: IOH and IP over time with warm-up period

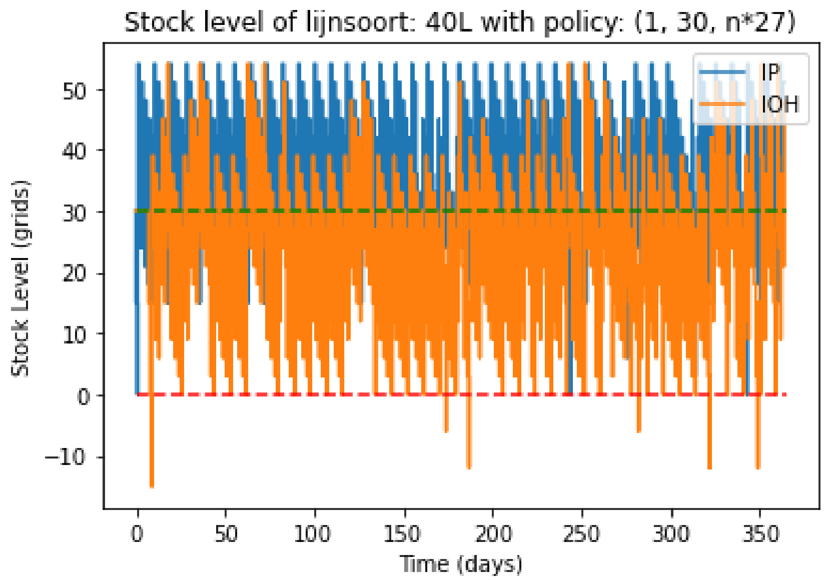


Figure 24: IOH and IP over time with warm-up period

In Appendix D it can be seen that reorder levels for all line types are determined. The review period varies between 1, 2 and 7 days. The lead time and demand have a low and high variant. When it is low, it is decreased with 20% and when it is high, it means the average is increased with 20%. This is done to simulate (non) busy periods within the simulation. The reorder levels are displayed in kached rolls. This means that they can be converted to grids or fiber containers by using the data from Section 3.1. These results are determined from the DES with a simulation length of 2.5 years. Each simulation was done 500 times which leads to accurate average results. The most realistic reorder levels are shown in Table 9. In this scenario, the exact historic demand is used, the lead time is average and the review period is set to one day. It shows that for the line types, 60L, 70L and 74L, the reorder levels are the larges. This can be explained because these are the line types with the largest demand. Line type 44L also has a relative large reorder level, because this line type has a very large waiting time included in the process.

Table 9: Reorder level (in kached rolls) of all line types with average demand, average lead time and $R = 1$

	31L	**36L**	**40L**	**41L**	**44L**	**50L**	**52L**	**57L**	**60L**	**67L**	**70L**	**74L**	**80L**	**85L**
s	12	8	11	11	17	3	5	7	28	10	25	28	13	10

The main results for the reorder levels are: (1) it increases when the Lead Time increases, (2) it increases when the review period becomes longer, and (3) it increases when the demand becomes larger. If the Lead Time becomes larger it means that it takes longer to supply the products. To ensure that the IOH does not become negative, the reorder level is increased. If the review period becomes longer, it takes more time to notice if the IP dropped below the reorder level and in this time period, the IOH may already be below zero. At last, if the demand increases, the reorder level increases. This also makes sense, since more products are required in the same time period. This will lead to a steeper decrease in IOH and therefore it is more likely that it will drop below zero. Therefore, the reorder level has to be increased to keep at least the 99% fill rate.

4.2 Dual Resource Constraint Flexible Job Shop Scheduling with Setup Operators

In this section, the results of solving the DRC-FJSSP-SSO will be discussed. In Section 4.2.1, the MILP will be reviewed. Section 4.2.2 elaborates on the results of the GA.

4.2.1 MILP

The MILP was solved with three different solvers: PulP's main solver, GUROBI and GLPK. From these three solvers, GUROBI performed far better than the other two. The computation time for scheduling six jobs using GUROBI was about 1 minute, while the others could not deal with six jobs within an hour. This is shown in Table 10. The large computation time is the reason why only the results of the GUROBI solver are used.

Table 10: Computation time in seconds for different solvers

#jobs \Solver	GUROBI	GLPK	PulP
4 jobs	4	>3600	1882
5 jobs	8	>3600	>3600
6 jobs	62	>3600	>3600

Figure 25 shows an optimal solution when processing six jobs. On the vertical axes, the machines are displayed. m2 and m3 are used for operation 1, m12 - m59 for operation 2 and m88 for the last operation. It can be seen that this schedule is optimal. This is because Job 4, the purple bar in Figure 25, has the longest processing time, but also starts first and finishes last. Because Job 4 is not able to produce quicker, this is the minimum time it takes to complete all six jobs. There might be a possibility that Job 4 can be processed on another machine to reduce the production time, but this is checked and is not the case. This means that for these six jobs the optimal production time is 404 minutes. As mentioned before, the solver took 62 seconds to output these results. If the amount of jobs is increased, the computation time also increases. Seven jobs already took 8 minutes and the solution for 8 jobs was not yet found after 2 hours. Computation time increases exponentially from here. To overcome the problem of extreme large computation times, a developed GA is used.

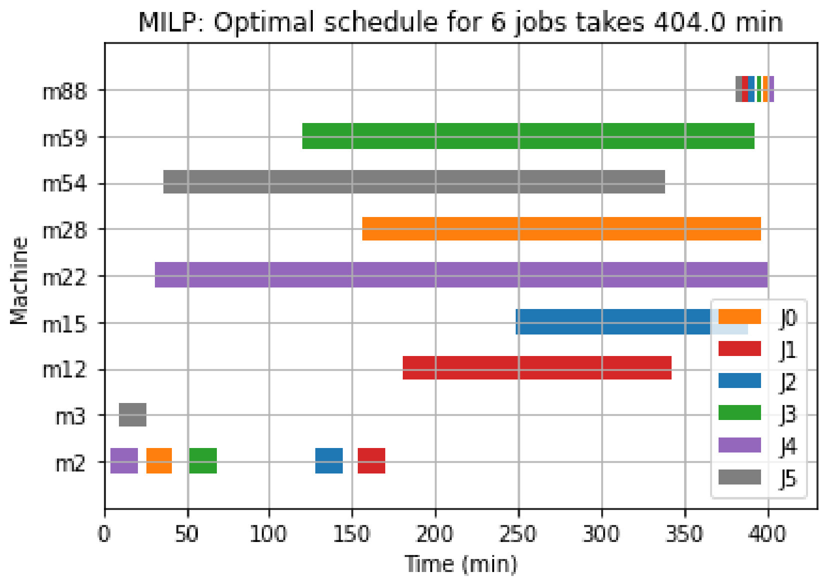


Figure 25: Optimal Solution for 6 jobs using the GUROBI LP solver

4.2.2 GA

First, it had to be decided which type of starting population would be used, a random or biased starting population. As mentioned in Section 3.2.2, three different starting populations with corresponding operators were examined. All were tested using "standard" hyper-parameter settings. The tournament size is usually set around 3, the crossover probability at 0.9 and the mutation probability at 0.05 Nobile (2021). These are however problem-specific, thus they will be optimized later on. The GA has ran 1000 generations and this was done 50 times. The average fitness values of the 50 runs can be seen in Figures 26, 27 and 28. A random starting population had on average a fitness value of 578582. The biased starting population with consecutively operations of the same job had a fitness value of 348981 and the biased starting population where first all operations of Operation 1, than 2 and last 3, were scheduled had on average a fitness value of 556242. Figure 28 also contains more variation within the different runs as can be seen by the large fluctuation in for the worst individual.

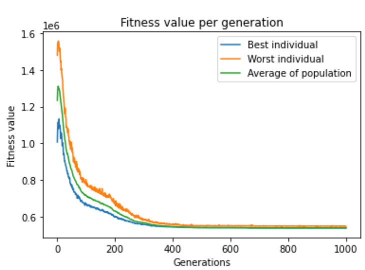


Figure 26: Random starting population

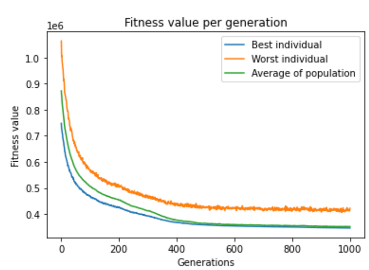


Figure 27: Biased starting population, (O1,O2,O3,O1,O2,O3)

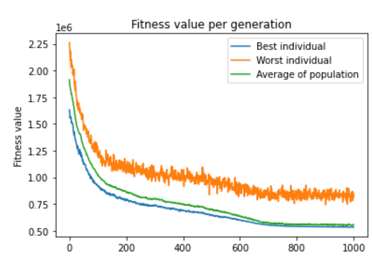


Figure 28: Biased starting population, (O1,O1,O2,O2,O3,O3)

It can be concluded that the biased starting population including the corresponding operators performed significantly better than the random starting population and operators. This means that all upcoming results will be using a biased starting population and biased operators. The bias that is added is that Operations of the same job are scheduled consecutively like in Figure 29.

Figure 29: Example of biased schedule that will be used

Next, the hyper-parameters (tournament size, crossover probability and mutation probability) were determined. The tournament size is usually set around 3, the crossover probability around 0.9 and the mutation probability around 0.05 Nobile (2021). However, this is problem-specific, which meant that different combinations have to be checked. In Appendix E the results of the different configurations can be seen. In all these situations that are checked there were 35 Jobs, one operator for Operation 1, three operators for Operation 2 and one operator for Operation 3. Every configuration was tested using 1000 generations which were ran 50 times. The average solution of these 50 runs at the end of 1000 generations is presented in the tables in Appendix E. It can be observed that $k = 3$, $P_c = 0.9$ and $P_m = 0.15$ results in the lowest fitness values. Therefore, these are the setting that will be used to obtain all results.

Finally, it is also concluded that the termination criterion is set to be a maximum number of generations. This is due to the fact that the GA should explore as long as possible. Because the mutation probability is relatively large, it can impact the population even when there is almost none variation left in the population.

MILP comparison
When using the same six jobs as for the MILP, the GA found the optimal solution of 404 minutes similar to the MILP, see Figure 30. This was done using a random starting population. Within six generations, an optimal solution was already found. Running six generations of this schedule takes less than 1 second. It can be concluded that it is faster to use the GA than the MILP solver.

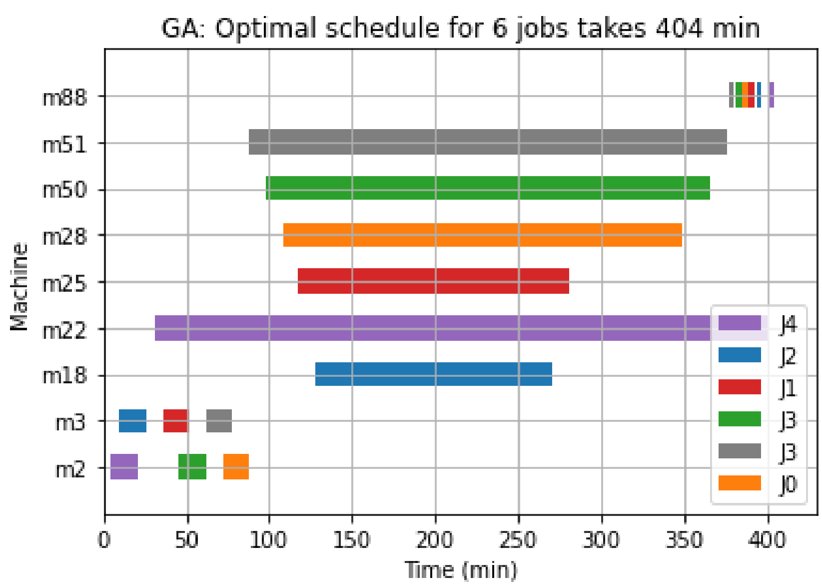


Figure 30: Optimal Solution

1 day schedule
When production at the factory is going well in stage 2 of the production line, 120-130 grids are being processed per day. This is based on internal data. This is the reason why for 1 day, 128 jobs

are scheduled in this GA. By doing this, the solution can be benchmarked. That means that the 128 jobs should be able to be processed within 24 hours.

The amount of generations that a GA runs, impacts the quality of the solution. If the GA runs for more generations, it will more likely lead to a better solution. The GA tries to minimize the completion times of every operation. The fitness value of the solutions will slowly decrease over time, but after a certain amount of generations, the fitness will not decrease as fast anymore. From this point, the mutation is required to receive better schedules, because more and more individuals in the population become similar. This can be seen in a convergence plot in Figure 31. This Figure shows the average fitness value per generation of 50 different runs. This means, the same GA has ran 50 times and the fitness values per generation are saved. Next, these average of the fitness value per generation was determined. In Figure 31, the minimum-, average- and maximum fitness values of each generation are visualized. At approximately generation 450, it can be seen that the decrease in fitness becomes smaller. The average population fitness and best individual fitness almost become similar. Based on this, it can be said that there is not much variation left in the population. However, the fitness value will still be decreasing over time. The mutation operator makes sure this happens. It changes small parts of a schedule to create diversity in the population.

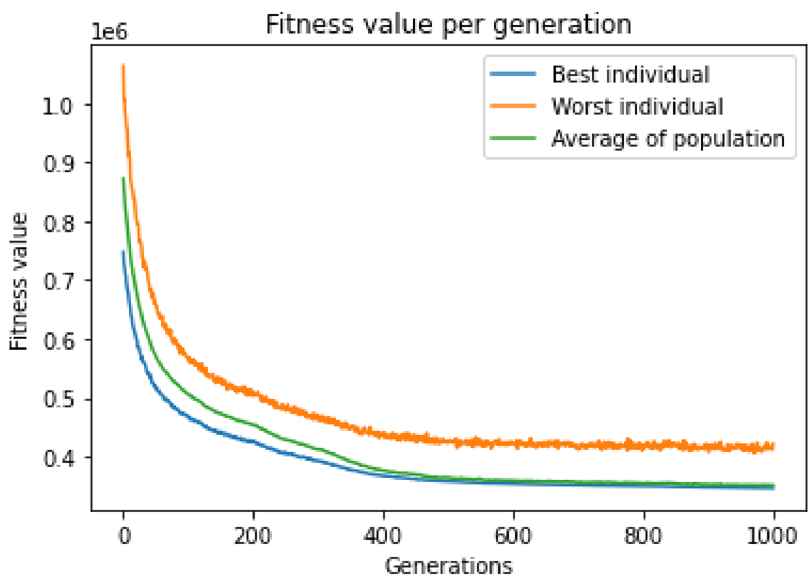


Figure 31: Convergence plot

In Figure 32 the best-found solution can be seen. For this schedule, one operator for Operation 1, three operators for Operation 2 and one operator for Operation 3 are required. The GA used 40000 generations, which took 603 minutes to run. Most jobs (108) end before 1440 minutes, which is 24 hours. This means that when this schedule is used, 128 jobs can be started in a day and this will lead to 128 jobs being finished within 24 hours. This is since at the end of the night certain machines are still active, but not yet finished. These jobs are then ready the next day. These are the jobs that finish after the 1440 minutes. This means when you take WIP into account, 128 jobs can be processed within 24 hours.

Figure 33 and 34 show the utilization of the operators and machines respectively. Figure 33 illustrates that the utilization of the operator of Operation 1 (OMM1) is 62% of the time busy with setting up machines. Operators 1, 2 and 3 (OPP1, OPP2 and OPP3) for Operation 2 have on average a utilization of 45%. At last, the operators which performs Operation 3 (OML1) has a utilization of 7%. When checking the utilization of the machines, it can be concluded that machine

2 and 3 are mostly used during Operation 1. Next, machines which use the STB method in Operation 2, have on average a utilization 55%. The other machines within Operation 2 have on average utilization 43%. Machine 88, which performs Operation 3, has a utilization of 28%. Figure 34 also shows that some machines are used more often than others. The reasons for certain machines being more used than others are (1) the demand for grids which can be produced on this machine is larger and (2) the processing times for certain machines differ. This means that some machines can process a grid faster than others.

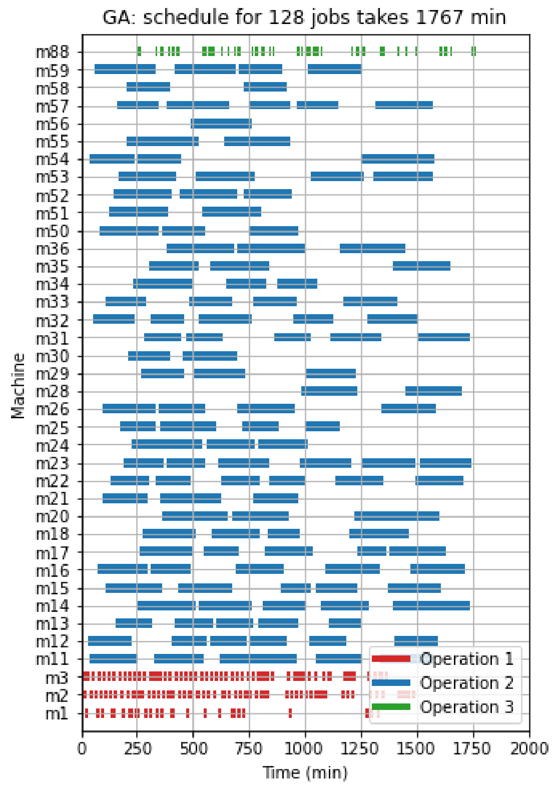


Figure 32: Schedule after 40000 generations

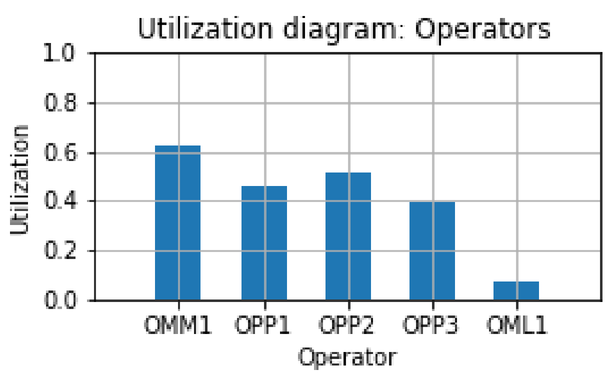


Figure 33: Operator utilization 128 jobs

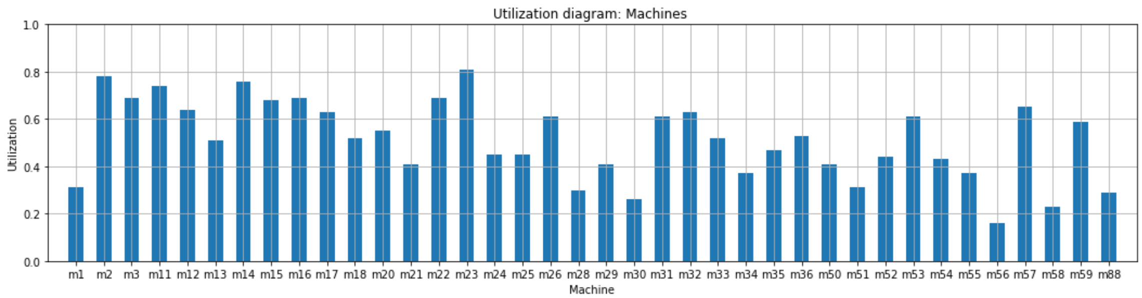


Figure 34: Machine Utilization 128 jobs

As can be seen in Figure 32, three operators seem necessary to deal with all the jobs in Operation 2. To check this, the number of operators is changed for Operation 2 in steps of 1 form 2 to 6. Average schedules of these different schedules can be seen Figures 45, 46,47 and 48 in Appendix F. All five schedules were running for 1000 generations and this was done 50 times. As illustrated, 2 operators at Operation 2 resulted on average in a makespan of 2200 minutes, 3 Operators lead to a schedule around 2100 minutes, the GA generated a schedule with 4 Operators to be finished in 2000 minutes, 5 operators were on average finished within 1960 minutes and 6 operators also required around 1960 minutes. However, the makespan is not minimized. The completion times of every operation are minimized. This was done to make sure the GA kept decreasing its fitness value. Besides, minimizing the total completion times of all operations will also lead to low makespan(Maassen et al., 2020). In Appendix F, the convergence plots can be seen of each type with corresponding fitness values. It shows that the GA with 2 operators leads to an average fitness value of 365243, the GA with 3 operators to a fitness value of 348981, the GA with 4 operators to a fitness value of 337125 and when 5 or 6 operators are used, the fitness value was on average 333413 and 333386. Having 5 operators could therefore be beneficial. 6 Operators would not further improve the schedule. A remark is that this cannot be stated for certain, because the schedules are not optimal.

The utilizations of each operator and all machines are also shown in Appendix F. For the operators, the results show that the utilization of Operation 1 and 3 are respectively 60% and 5%. The utilization of operators of Operation 2 were respectively 60%, 40%, 30%, 25% and 22% for a GA with 2-, 3-, 4-, 5- and 6 operators. The machine utilization, Appendix F, did not give different results when changing the number of operators. The average utilization of the machines was 43%. It is visible that some machines are more often used than others, for example, Machine 2 for Operation 1 and Machines 12, 17, 32 and 36 for operation 2. On the other hand, Machines 13 and 33 were used less compared to the other machines. It can also be seen that the machines which have used the LTB method during Operation 2, have on average a smaller utilization than machines that use the STB method. This can be explained by the larger demand for grids that use the STB method.

The utilization diagrams of Figures 32 and 46 can be compared. It can be seen that the utilization of the schedule which has ran 40000 generations is almost everywhere larger. This is due to the fact that this schedule is more optimal than the other and therefore, the operations are scheduled with less waiting time.

3 day schedule

Running a schedule that consists of 350 jobs (which is a normal production target for three days) takes even longer than for 128 jobs. Running 1000 generations for a schedule that contains 350 jobs takes about 35 minutes to finish, while it only takes 13 minutes for a schedule with 128 jobs. This is because when increasing the number of jobs, the schedule becomes larger and larger. This makes it more complex for the GA to find better solutions. However, this was known beforehand and the large complexity was the reason why a GA was chosen to solve the problem. Figure 35 illustrates a solution that has run 20000 generations and uses one operator for Operation 1, three for operation 2 and one for operation 3. The makespan of this schedule is 4673 minutes. As mentioned, 350 jobs are usually produced within three days. Three days contain 4320 minutes, this means this schedule almost competes with the current schedules at Philips, because almost all jobs are processed in 3 days.

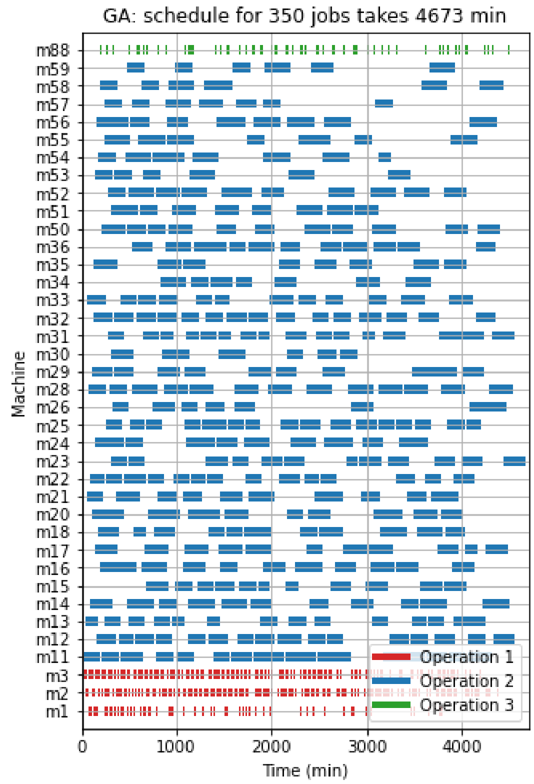


Figure 35: Schedule after 20000 generation of 350 jobs

When comparing the machine- and operator utilization of a 350 job schedule to a 128 job schedule with 3 operators in Operation 2, it is visualized that the utilization for the operators is about 10% larger, see Figure 36. This can mean that this schedule is even better than the 128 jobs schedule. Figure 37, which shows the machine utilization, illustrates the same patterns as before. It is also visible that the utilization increased by about 10%. The most used machine is still machine 2. Due to the fact that none of the machines have a large utilization, a bottleneck cannot be determined.

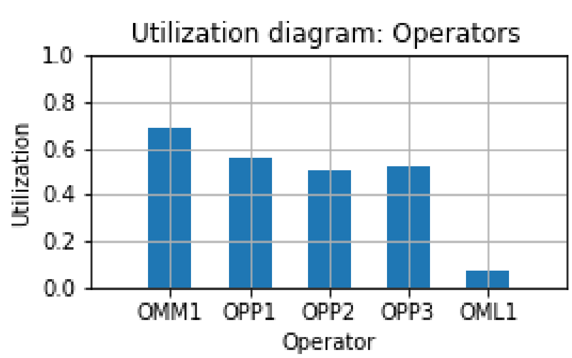


Figure 36: Operator utilization of 350 Jobs

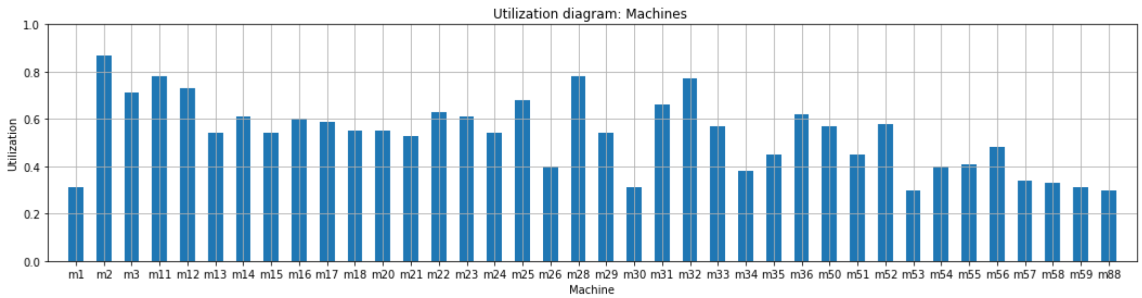


Figure 37: Machine utilization of 350 Jobs

Another insight from the 350 job schedule, is the number of jobs that can be processed within one day. The GA with 128 jobs proved that it when taking WIP into account, can handle 128 jobs in a day. The maximum amount of jobs that were finished in 24 hours in the 350 job schedule is 144 for Operation 1, 148 for Operation 2 and 153 for Operation 3, see Figure 38. This means that if there is one operator available for Operation 1, three for operation 2 and one for Operation 3 for 24 hours, it is possible to finish around 150 jobs. This shows that Figure 32 is far from optimal.

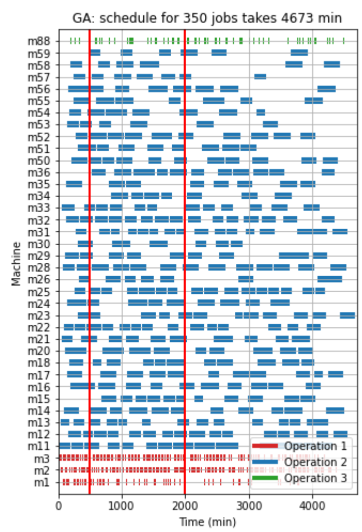


Figure 38: 150 jobs end in 24 hours

Next, the type of job processed on a machine or by an operator is examined. First, every job where an operator performs a setup is visualized in Figure 59 in Appendix G. As can be seen in the figure, the setups that every operator performs are randomly spread over the schedule. This means that every operator performs setups for all machines and that operators do not need to be assigned to specific machines. Second, the amount of lamellae on a grid is examined. The more lamellae a grid contains, the longer the processing time takes during Operation 2. What can be concluded from Figure 60, is that during the first 2000 minutes of the schedule, significantly more grids are processed with a lower amount of lamellae (<1700). For the medium size grids (number of lamellae between 1700 and 2200) it is observed that they are more likely to be produced at the end of a schedule. The large grids with more than 2200 lamellae do not show any pattern and are produced evenly spread out over the schedule. The percentages of grids that are produced in a certain time period with a low, medium or a large number of lamellae are shown in Table 11. For example, between minute 0 and 500, 46% of all processed jobs have less than 1700 lamellae, 21% have between 1700 and 2200 lamellae and 33% have more than 2200 lamellae. At last, grids have to characteristics which are important fot setups, height and line type. Figure 61 in Appendix G visualizes grids with a similar height/line type combination. This combination determines how long the setup will take. There are five colors which represent the most common combinations. Red, orange, purple, blue and grey represent respectively (1.5, 41L), (1.5, 60L), (1.5, 70L), (1.5, 74L) and (1.75, 40L). The first number represents the height and the second number the line type. Green represents the remaining grids. Figure 61 in Appendix G shows that it occurs often that two or more jobs with the same height/line type combination are processed consecutively on a machine. This can be explained by the fact that processing similar jobs after one another reduces setup time. However, this does not occur that often. When only the considering the height of a grid, it occurs more often these grids are processed after one another. This makes sense, because a difference in height has a significant larger impact on the setup time than a line type difference. Because this is also not the case for all of the operations, it probably is the case that the schedule is not optimal as expected.

Table 11: Production percentage per time period based on number of lamellae

lamellae \min	0-500	500-1000	1000-1500	1500-2000	2000-2500	2500-3000	3000-3500	3500 -4000	4000-4500	4500-5000
<1700	0.46	0.43	0.34	0.41	0.32	0.24	0.25	0.16	0.22	0
>1700 & <2200	0.21	0.30	0.30	0.37	0.36	0.55	0.47	0.61	0.61	1
>2200	0.33	0.27	0.36	0.22	0.32	0.21	0.28	0.23	0.17	0

4.3 Capacity Scheduling

First, the activities which are performed by an operator per processing step were determined together with the time it takes to perform these activities. These activities with corresponding times can be found in Appendix H. With these processing times a happy flow can be determined for all three types of grids, Regular-, Mammography- and Round grids, Table 12. It is assumed that there are no additional waiting times and that the grid can be shipped directly to the painting shop. However, this is not likely, because grids can only be shipped to the painting shop at one moment during the week.

Table 12: Happy flow cycle time stage 3

Regular	Mammography	Round
9.04 days	10.25 days	11.06 days

The next step is to determine which process step is the bottleneck in stage 3. This depends on two resources: Operators and Machines. Three processes can lead to bottlenecks, which are Post-processing 1, Veneer and Post-processing 2. For these three processes, it is shown how many operators are available per shift in Appendix I. The configuration 1-1-1 means that in every shift one operator is available. For example, on line 2 in Appendix I, the "Post-processing 1" process has one operator for every shift. With this amount of operators, 110 grids can be processed in 24 hours. For the process step "Veneer" 115 grids can be processed, etc. By creating a lot of different scenarios, the bottleneck of stage 3 can be determined for all of those. For example, the bottleneck for line 2 is Post-processing 1 and it has a bottleneck rate of 110 grids per day.

4.4 Combination

Combining all three stages gives the following results. First, if the reorder levels and review periods of Appendix D are used it can be said that there will be at least a 99% certainty that there is enough inventory to start producing at stage 2. In stage 2, 128 jobs are scheduled per day with one operator for Operation 1, three for Operation 2 and one for Operation 3, see Figure 39. Stage 3 is added to the GA and this results in Figure 40. This figure visualises the start- and end time of each job for Post-processing. Subsequently, it shows how many jobs are processed over time. This differs, because during the quality checks, some of the grids are scrapped. At quality check1, on average 21,2% of the grids are discarded and during quality check 2, 23% of the grids are scrapped. The first decline that can be seen in Figure 40 represents all grids that did not pass quality check 1. The second decline in the figure are all the grids which did not pass quality check 2. The last decline are all the grids which are completes and send to the customer. It can be concluded that in between quality check 1 and quality check 2 around 105 grids were processed. This means that in total 105 grids have to be processed in stage 3 to keep up with the production in stage 2. This is possible with the right set of operators, which can be found in Appendix I. A set of operators have to be found which have a bottleneck rate larger than 105 grids per day. Finally, in total around 77 grids were completed.

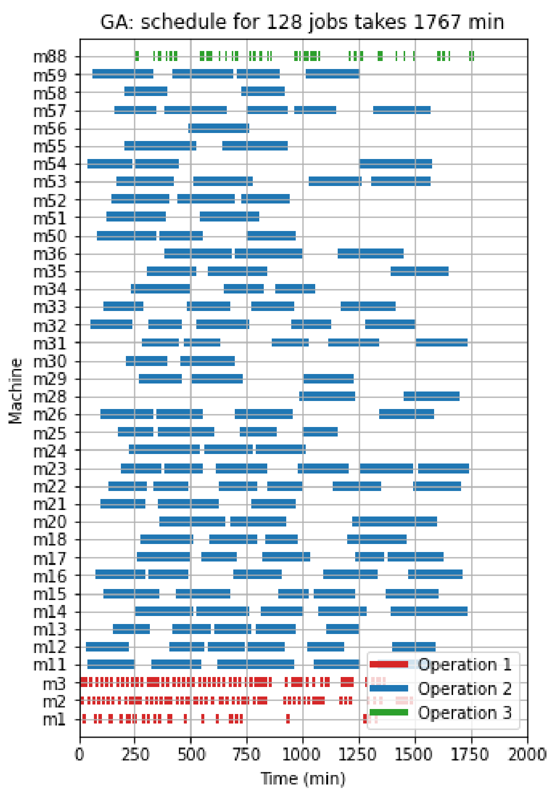


Figure 39: Schedule stage 2

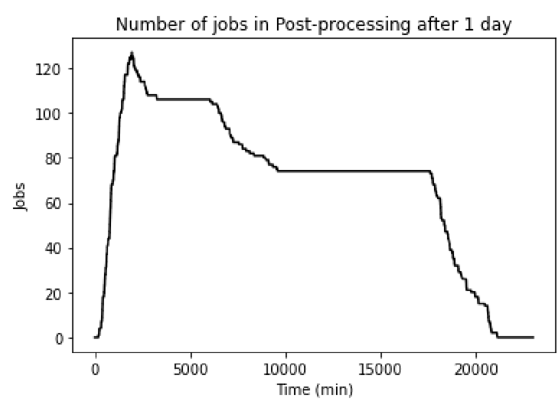


Figure 40: Number of jobs in stage 3 from 1 day of production in stage 2

Next, the average total production time can be determined for the production line and be compared to the current average production time. The four different processing times that are required are, (1)the processing times of stage 2, (2) the time from quality check 1 to quality check 2, (3) the time it takes from quality check to sending the grid and finally (4), the time it takes to start a new job once it is scrapped during a quality check. The average processing time for a random job on a day in stage 2 is 0.5 day. This is not optimized in this thesis. The average processing time between quality check 1 and quality check 2 was 8.32 days. When having enough operators to keep up with all demand, this should be possible in 5.03 days. This is based on the time it takes for a grid to go through all activities in Appendix H. Next, the time between quality check 2 and sending the order to the customer is on average 10.5 days. This cannot be improved, due to the fact that most of the time the grid is at an external location. Finally, the time it takes to start a new production for a certain order can be improved. Currently, it takes about half a day to start a new production. This is due to the fact that new jobs are released twice a day. When this is done directly, this can save up to 0.4 day per grid.

Combining this with the knowledge of the yield, Figure 22 can be entered. For this Markov chain the steady state of the time it takes to complete can be determined. First, the current average total production time is determined, see Figure 41. On average, it takes 22.4 days to complete a grid, but the time it takes to complete 90% of the grids is already 29.6 days. With the new processing times per processing step, the average total production time is equal to 17.9 days and 90% of these jobs will be finished after 22.3 days. This can be seen from Figure 42. Both simulations were ran 10 million times.

When having the right amount of operators at the right section in the production line can significantly improve the total production time. Next to that, by decreasing the time when a grid is scrapped until a new production is started the average total production time also decreases by 0.2 days. This has, as can be seen, as can be seen a smaller impact.

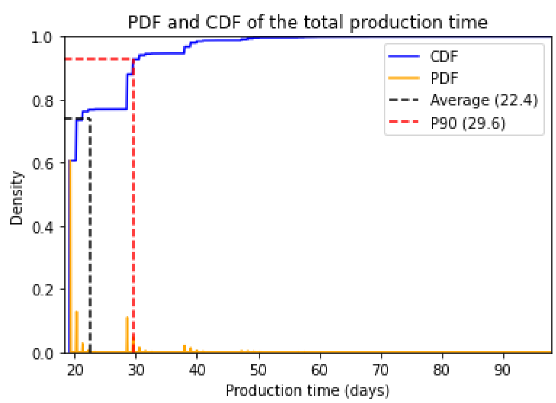


Figure 41: Current total production time

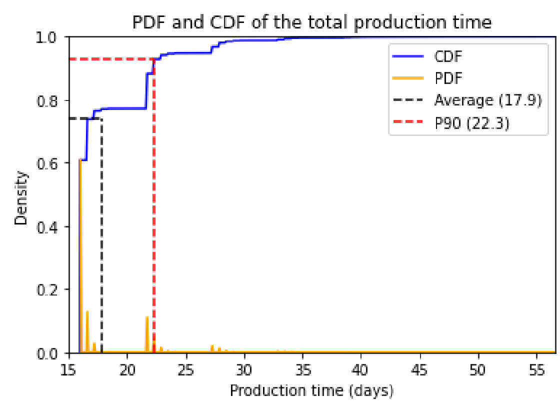


Figure 42: Improved total production time

5 Conclusion & Discussion

In this chapter, first, the research question will be discussed in Section 5.1. Next, Section 5.2 describes the limitations of this thesis. Subsequently, Section 5.3 provides recommendations for Philips. Lastly, Section 5.4 will discuss possible directions for future research.

5.1 Conclusion

In the studied multi-stage manufacturing system of Philips, anti-scatter grids are produced from raw materials. To ensure customers receive the anti-scatter grids before the RDD while considering constraints like yield, capacity and growing demand, planning and scheduling have to be on point. Therefore, this thesis investigated planning and scheduling improvements for the multi-stage manufacturing system. The main research question is as follows:

"How should Philips plan capacity and schedule orders to improve the OTTR, while keeping in mind the constraints?"

To answer this research question, first, each stage had to be optimized. In stage 1, a discrete event simulation has been developed that determines the best inventory policy which guarantees a 99% fill rate. For each line type, a different inventory policy was generated and these can be renewed by updating the historic demand data. By implementing the generated inventory policies, a 99% fill rate is guaranteed. Therefore, it can be assumed that stage 1 is optimized as there is always enough inventory to start the production process in stage 2.

Stage 2 can be described as a DRC-FJSSP-SSO. In literature, these problems are most of the time solved using a GA, because exact solutions use too much computation time. This was checked by creating a MILP model and testing several instances. An exact solution method indeed used too much computation time. Therefore, a GA was developed which takes machines, operators and variable setup times into account. Since every problem is different, several genetic algorithm operators are tested to check which performs best for this type of problem. It is concluded that a biased population, which schedules operations of each job right after the other, performed best based in the fitness function. With these bias, a JOSO crossover operator and a machine/operator mutation were developed. Next, the hyper-parameter settings also had to be optimized for this specific problem. The optimal settings are (1) the tournament size is set to 3, (2) the crossover probability is set to 0.9 and (3) the mutation probability is set to 0.15. With the right settings for this problem type, a schedule for stage 2 can be generated. However, the GA did not outperform the current scheduling method. The GA could generate similar schedules as currently, but could not improve them due to the complexity of the problem. However, it can be concluded that having more than five operators in Operation 2 is not beneficial and the setup time is minimized when producing grids with similar heights on the same machine consecutively. Furthermore, at least 150 jobs can be processed within 24 hours using 3 operators at Operation 2. This means, that when taking yield into account, at least 488 grids can be completed within a week. Currently, 400 grids are completed per week.

Knowing that around 120 jobs are processed every day in stage 2, the average number of jobs to be processed in stage 3 can be determined. The average yield in quality check 1 is equal to 0.787, which means that on average 95 jobs have to be processed per day in stage 3 to keep up with incoming grids. To meet this requirement, the right amount of skilled operators have to be allocated to each each processing step. A thorough analysis is performed to see how long operators perform activities at each processing step. With this knowledge, the number of operators can be determined at each processing step to process at least 95 grids per day. Moreover, it can be concluded that the duration of the happy flow of the complete production line for regular grids is 9.04 days in stage 3. In this happy flow, no waiting time is taken into account which is not very realistic.

The GA of stage 2 can be revised in a way that takes stage 3 also into account. This is done by adding an extra operation. The same genetic algorithm operators and hyper-parameter settings are used. When running the model, a schedule for stage 2 is still generated, but also the inventory of stage 3 is visualized. When the grids are finished in stage 2, the grid starts production in stage 3. Using the average yield, some grids are scrapped within stage 3 and the inventory decreases again. This figure shows how many jobs have to be processed in stage 3 and with this knowledge the right number of eligible operators can be determined to handle this flow. Furthermore, it compares the total production time in the current and new situations. This is done by creating a Markov chain and determining the steady-state of the total production time. 90% of all grids are currently processed (stage 2 and stage 3) within 29.6 days, while in the new situation it only takes 22.3 days to accomplish this. This means, that when processing around 120 grids in stage 2, it is possible to deliver 90% of all grids within 22.3 days if the correct number of operators with the right skill-set are available. This is a decrease of 24.6%. For this scenario, 2 operators should be available for shifts 1 and 2 at Post-processing 1 and Post-processing 2 and 1 operator should be working every shift to Veneer. This will lead to a total production time of 22.3 days and therefore it will be possible to meet the RDD if the production is started in time.

5.2 Limitations

In this thesis, several assumptions were made to simplify the production environment. These assumptions influence the results. For example the processing times of each production step. Most processing times are not recorded by Philips. In a few cases, there is some data available, but most often, the only way to define a processing time is by making an educated guess based on experienced operators. It cannot be checked if these are reliable processing times, because the right data is not registered. Another assumption that is made is that machines are always active and do not break down. This is not realistic. Most of the machines are becoming old and maintenance is required more and more often. For some processes, a machine can break down multiple times per order handled. This will severely impact the processing times.

Another limitation of this research is that the schedules that are generated by the GA are not optimal. Because solving this type of problem is very complex, the computation time is still very large. It is much better than solving it using an LP solver, but when running the GA for 24 hours it still does not find an optimal solution. Furthermore, the GA can be very random with its solutions. The first time it might perform very well and the second time it can be the case that it performs poorly. That is one of the disadvantages of using a GA.

The final limitation is regarding the fitness function of the GA. Currently, the completion time of each operation is minimized while minimizing the makespan is the goal. The makespan is not chosen to be the fitness function, because this led to more computation time and worse schedules within a given time period. It can be argued that minimizing the total completion time lowers the makespan (Maassen et al., 2020), however, when minimizing the completion time of each operation it cannot be stated with 100% certainty that the makespan is always minimized. In some cases, the fitness value was very low, but the makespan was large.

5.3 Recommendations

There are several recommendations regarding planning and scheduling to decrease the total production time and with that increase the OTTR. For stage 1, Philips should take new reorder levels into account to make sure that there is always enough stock to start producing in stage 2. The review period should be one day to decrease the reorder levels. Next, the inventory policies should be updated once every half year, or once the demand pattern changes drastically. By doing this, the inventory policy is taking the current demand pattern into account.

For stage 2, three operators are required for the fusing step in the production line. With three operators, 150 grids per day can be processed and this is enough to meet the goal of 400 completed grids per week. The grids should be scheduled based on the characteristic "height", because the setup times will be minimized when multiple grids with the same height are processed consecutively on the same machine.

Finally, the number of operators at each processing step in stage 3 should be based on how many grids are produced in stage 2 of the system. The bottleneck rate of stage 3 should at least be larger than the number of grids that are produced in stage 2 minus the yield. By doing this, the WIP will not be able to increase and therefore, a long cycle time and a worse yield can be prevented. Due to the fact that Post-processing 1, Veneer and Post-processing 2 are the steps where the bottleneck is most likely, it might also be useful to train more operators to perform these activities. These steps are crucial within the total production line and it is, therefore, harder to process piled-up WIP. Once a grid is scrapped at a quality check it can save time to directly release a new production order instead of waiting for the next day.

When keeping all these improvements in mind, the total production time will decrease and it will be possible to guarantee a 6-week delivery period. There is only one problem. Currently, there is a large backlog which means that products are not produced right away. If this takes longer than two weeks, less than 90% of the grids can be delivered within six weeks. This is because 29.6 days are required to complete 90% of the jobs. Therefore, another recommendation would be to spend a couple of weeks with the maximum number of operators in every shift to reduce this backlog.

5.4 Future Research

The main limitation of this research was that the solution of the GA was not improving the current scheduling method. Future research is required to decrease computation time or increase the effectiveness of the GA. There are two options for this, (1) the GA is revised and (2) a new method is developed. For option (1), the bias can be changed. A different bias might lead to better results. Next, the operators can be adjusted. By discovering new operators, the individuals might have better fitness and the computation time might decrease. Another possibility is to revise the fitness function. The fitness function has a large impact on the GA since this is optimized. Developing a better fitness function might lead to the GA converging faster to the global optimum. Option (2) might also be beneficial if the GA does not perform as hoped. For example, a heuristic will decrease the computation time, but the fitness value might not be as good. However, this is still something to consider when continuing with this thesis.

Another way to improve the model is by implementing downtime of all machines. This is done to make the model more realistic. Future research on how downtime can be implemented into the simulation and GA is therefore required. Because the simulation is a DES, this can be easily implemented. The only requirement is that the (average) time until machine failure occurs, has to be known. Currently, this is not known at Philips and therefore, much data has to be gathered over the upcoming period. Chaudhry and Khan (2016) mentions that an FJSSP, with taking downtime into account, is already developed. This can be a starting point for implementing downtime in this model.

Future research can also be done into finding a suitable fitness function that minimizes makespan. This can be done by proving to what degree minimizing the completion time minimizes the makespan, or by changing the fitness function for something that better minimizes the makespan. Proving to what degree minimizing the completion time minimizes the makespan can be useful in order to decide whether its is the right objective function or if it should be changed.

Finally, processing times have to be determined in more detail, because this affects the models and it is currently based on an educated guess. This means that future research has to be done into how data of thecomplete production process can be gathered and stored.

References

Abo-Zahhad, M., Ahmed, S. M., Sabor, N., and Sasaki, S. (2014). A new energy-efficient adaptive clustering protocol based on genetic algorithm for improving the lifetime and the stable period of wireless sensor networks. *International Journal of Energy, Information and Communications*, 5(3):47–72.

Anderson, D. R., Sweeney, D. J., Williams, T. A., Camm, J. D., and Cochran, J. J. (2018). *An introduction to management science: quantitative approach*. Cengage learning.

Applegate, D. and Cook, W. (1991). A computational study of the job-shop scheduling problem. *ORSA Journal on computing*, 3(2):149–156.

Azizi, N., Zolfaghari, S., and Liang, M. (2010). Modeling job rotation in manufacturing systems: The study of employee's boredom and skill variations. *International Journal of Production Economics*, 123(1):69–85.

Ballard, G. (2001). Cycle time reduction in home building. In *Proceedings of the 9th Annual Conference of the International Group for lean construction*, pages 06–08. sn.

Bokhorst, J. A. and Gaalman, G. J. (2009). Cross-training workers in dual resource constrained systems with heterogeneous processing times. *International Journal of Production Research*, 47(22):6333–6356.

Bon, A. T. and Samsudin, S. N. A. (2018). Productivity improvement in assembly line by reduction cycle time using time study at automotive manufacturer. In *Proceedings of the International Conference on Industrial Engineering and Operations Management Bandung, Indonesia, March*, pages 6–8.

Bowman, E. H. (1959). The schedule-sequencing problem. *Operations research*, 7(5):621–624.

Cachon, G. and Terwiesch, C. (2008). *Matching supply with demand*. McGraw-Hill Publishing.

Carlier, J. and Pinson, É. (1989). An algorithm for solving the job-shop problem. *Management science*, 35(2):164–176.

Cattin, P. (2016). Basics of x-ray.

Cellura, M., Ardente, F., Longo, S., Marsala, G., and Pucci, M. (2011). The use of genetic algorithms to solve the allocation problems in the life cycle assessment. In *Ecosystem Protection in a Sustainable World: A Challenge for Science and Regulation*. Setac Europe Office.

Chaudhry, I. A. and Drake, P. R. (2009). Minimizing total tardiness for the machine scheduling and worker assignment problems in identical parallel machines using genetic algorithms. *The International Journal of Advanced Manufacturing Technology*, 42(5-6):581.

Chaudhry, I. A. and Khan, A. A. (2016). A research survey: review of flexible job shop scheduling techniques. *International Transactions in Operational Research*, 23(3):551–591.

Coello, C. A. C., Pulido, G. T., and Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 8(3):256–279.

Defersha, F. M. and Chen, M. (2010). A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups. *The international journal of advanced manufacturing technology*, 49(1):263–279.

Díaz, H., González-Rodríguez, I., Palacios, J. J., Díaz, I., and Vela, C. R. (2020). A genetic approach to the job shop scheduling problem with interval uncertainty. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 663–676. Springer.

Eberhart, R. and Kennedy, J. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pages 1942–1948. Citeseer.

Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., and Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175.

Gel, E. S., Hopp, W. J., and Van Oyen, M. P. (2007). Hierarchical cross-training in work-in-process-constrained systems. *Iie Transactions*, 39(2):125–143.

Greeff, G. and Ghoshal, R. (2004). *Practical E-manufacturing and supply chain management*. Elsevier.

Holland, J. H. et al. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Hopp, W. J. and Spearman, M. L. (2011). *Factory physics*. Waveland Press.

Hottenstein, M. P. and Bowman, S. A. (1998). Cross-training and worker flexibility: A review of drc system research. *The Journal of High Technology Management Research*, 9(2):157–174.

Jamili, A. (2016). Robust job shop scheduling problem: Mathematical models, exact and heuristic algorithms. *Expert systems with Applications*, 55:341–350.

Kacem, I., Hammadi, S., and Borne, P. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(1):1–13.

Katherasan, D., Elias, J. V., Sathiya, P., and Haq, A. N. (2014). Simulation and parameter optimization of flux cored arc welding using artificial neural network and particle swarm optimization algorithm. *Journal of Intelligent Manufacturing*, 25(1):67–76.

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

Krajewski, L. J., Ritzman, L. P., and Malhotra, M. K. (2010). *Operations management: Processes and supply chains*. Pearson New Jersey.

Ku, W.-Y. and Beck, J. C. (2016). Mixed integer programming models for job shop scheduling: A computational analysis. *Computers & Operations Research*, 73:165–173.

Kulkarni, K. and Venkateswaran, J. (2015). Hybrid approach using simulation-based optimisation for job shop scheduling problems. *Journal of Simulation*, 9(4):312–324.

Lee, K.-M., Yamakawa, T., and Lee, K.-M. (1998). A genetic algorithm for general machine scheduling problems. In *1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No. 98EX111)*, volume 2, pages 60–66. IEEE.

Lenstra, J. K., Kan, A. R., and Brucker, P. (1977). Complexity of machine scheduling problems. In *Annals of discrete mathematics*, volume 1, pages 343–362. Elsevier.

Li, J., Sun, S., and Huang, Y. (2011). Adaptive hybrid ant colony optimization for solving dual resource constrained job shop scheduling problem. *J. Softw.*, 6(4):584–594.

Little, J. D. (1961). A proof for the queuing formula: L= λ w. *Operations research*, 9(3):383–387.

Maassen, K., Perez-Gonzalez, P., and Günther, L. C. (2020). Relationship between common objective functions, idle time and waiting time in permutation flow shop scheduling. *Computers & Operations Research*, 121:104965.

Manne, A. S. (1960). On the job-shop scheduling problem. *Operations research*, 8(2):219–223.

Mohammed, M. A., Khanapi, M., Ghani, A., Obaid, O., and Mostafa, S. (2017). A review of genetic algorithm application in examination timetabling problem. *Journal of Engineering and Applied Sciences*, 12(20):5166–5181.

Mourtzis, D. (2020). Simulation in the design and operation of manufacturing systems: state of the art and new trends. *International Journal of Production Research*, 58(7):1927–1949.

Nahmias, S. and Cheng, Y. (2009). *Production and operations analysis*, volume 6. McGraw-hill New York.

Nobile, M. (2021). 1bm120 – computational intelligence lecture 1: Introduction to the course / evolutionary computation.

Obimuyiwa, D. (2020). *Solving Flexible Job Shop Scheduling Problem in the Presence of Limited Number of Skilled Cross-trained Setup Operators*. PhD thesis.

Obimuyiwa, D. and Defersha, F. (2020). A genetic algorithm for flexible job shop scheduling problem with scarce cross trained setup operators. In *International Conference of Production Research–Americas*, pages 142–155. Springer.

Paksi, A. and Ma'ruf, A. (2016). Flexible job-shop scheduling with dual-resource constraints to minimize tardiness using genetic algorithm. In *IOP Conference Series: Materials Science and Engineering*, volume 114, page 012060. IOP Publishing.

Pezzella, F., Morganti, G., and Ciaschetti, G. (2008). A genetic algorithm for the flexible job-shop scheduling problem. *Computers & operations research*, 35(10):3202–3212.

Philips (2021). Annual report 2020.

Reyes, J., Aldas, D., Alvarez, K., García, M., and Ruíz, M. (2017). The factory physics for the scheduling: Application to footwear industry. In *SIMULTECH*, pages 248–254.

Sani, B. and Kingsman, B. G. (1997). Selecting the best periodic inventory control and demand forecasting methods for low demand items. *Journal of the operational research society*, 48(7):700–713.

Sha, D. and Lin, H.-H. (2010). A multi-objective pso for job-shop scheduling problems. *Expert Systems with Applications*, 37(2):1065–1070.

Silver, E. A., Pyke, D. F., Peterson, R., et al. (1998). *Inventory management and production planning and scheduling*, volume 3. Wiley New York.

Singha, K., Buddhakulsomsiri, J., and Parthanadee, P. (2017). Mathematical model of inventory policy under limited storage space for continuous and periodic review policies with backlog and lost sales. *Mathematical Problems in Engineering*, 2017.

Sivanandam, S. and Deepa, S. (2008). Genetic algorithms. In *Introduction to genetic algorithms*, pages 15–37. Springer.

Smith, W. E. et al. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66.

Spanos, A. C., Ponis, S. T., Tatsiopoulos, I. P., Christou, I. T., and Rokou, E. (2014). A new hybrid parallel genetic algorithm for the job-shop scheduling problem. *International Transactions in Operational Research*, 21(3):479–499.

Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.

Tang, M., Gong, D., Liu, S., and Zhang, H. (2016). Applying multi-phase particle swarm optimization to solve bulk cargo port scheduling problem. *Advances in Production Engineering & Management*, 11(4).

Taylor, B. W., Bector, C., Bhatt, S., and Rosenbloom, E. S. (2013). *Introduction to management science*. Pearson.

Thürer, M. (2018). Dual resource constrained (drc) shops: Literature review and analysis. In *IFIP International Conference on Advances in Production Management Systems*, pages 81–86. Springer.

Treleven, M. (1989). A review of the dual resource constrained system research. *IIE transactions*, 21(3):279–287.

Treleven, M. D. and Elvers, D. A. (1985). An investigation of labor assignment rules in a dual-constrained job shop. *Journal of Operations Management*, 6(1):51–68.

van Donselaar, K. and Broekmeulen, R. (2017). Stochastic inventory models for a single item at a single location.

Wagner, H. M. (1959). An integer linear-programming model for machine scheduling. *Naval research logistics quarterly*, 6(2):131–140.

Werner, F. (2013). A survey of genetic algorithms for shop scheduling problems. *Heuristics: Theory and Applications*, pages 161–222.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342.

Wolsey, L. A. (2007). Mixed integer programming. *Wiley Encyclopedia of Computer Science and Engineering*, pages 1–10.

Wu, R., Li, Y., Guo, S., and Xu, W. (2018). Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm. *Advances in Mechanical Engineering*, 10(10):1687814018804096.

Xu, J., Xu, X., and Xie, S. (2011). Recent developments in dual resource constrained (drc) system research. *European Journal of Operational Research*, 215(2):309–318.

Yazdani, M., Zandieh, M., Tavakkoli-Moghaddam, R., and Jolai, F. (2015). Two meta-heuristic algorithms for the dual-resource constrained flexible job-shop scheduling problem. *Scientia Iranica*, 22(3):1242–1257.

Yinan, Q., Tang, M., and Zhang, M. (2014). Mass customization in flat organization: The mediating role of supply chain planning and corporation coordination. *Journal of Applied Research and Technology*, 12(2):171–181.

Zhang, C. Y., Li, P., Rao, Y., and Guan, Z. (2008). A very fast ts/sa algorithm for the job shop scheduling problem. *Computers & Operations Research*, 35(1):282–294.

Zhang, J., Ding, G., Zou, Y., Qin, S., and Fu, J. (2019). Review of job shop scheduling research and its new perspectives under industry 4.0. *Journal of Intelligent Manufacturing*, 30.

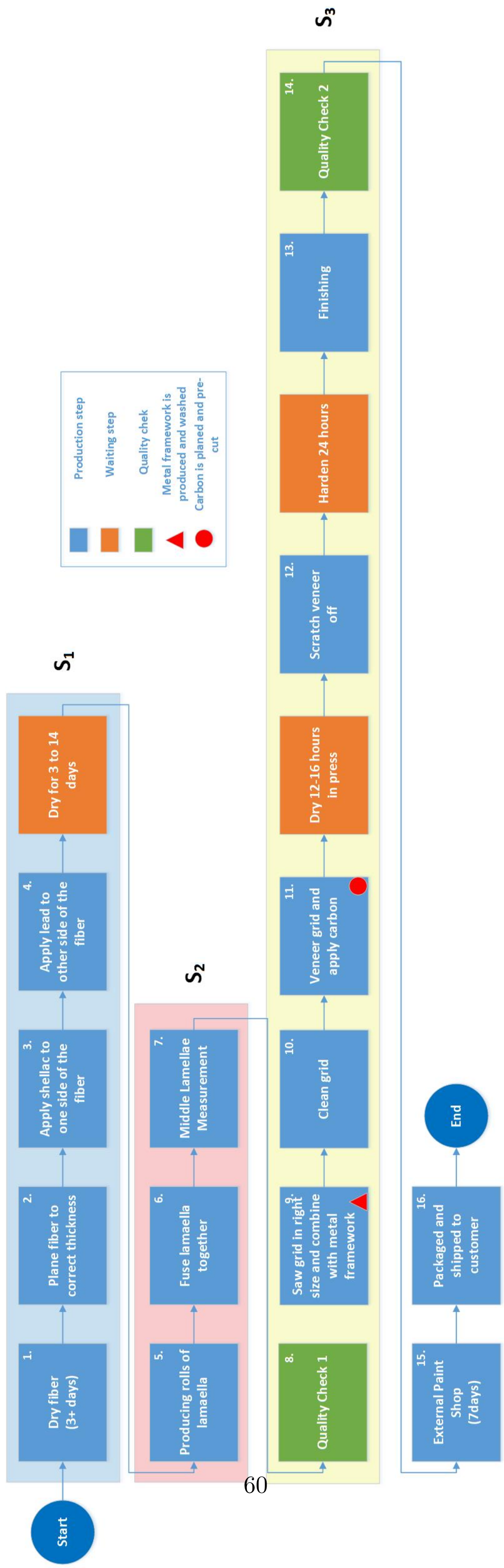
Zhang, J., Wang, W., and Xu, X. (2017). A hybrid discrete particle swarm optimization for dual-resource constrained job shop scheduling with resource flexibility. *Journal of intelligent Manufacturing*, 28(8):1961–1972.

Zhang, J. and Wang, X. (2016). Multi-agent-based hierarchical collaborative scheduling in re-entrant manufacturing systems. *International Journal of Production Research*, 54(23):7043–7059.

Zhong, Q., Yang, H., and Tang, T. (2018). Optimization algorithm simulation for dual-resource constrained job-shop scheduling. *International Journal of Simulation Modelling*, 17(1):147–158.

Appendices

A Process Flow



B Simplified Process Flow

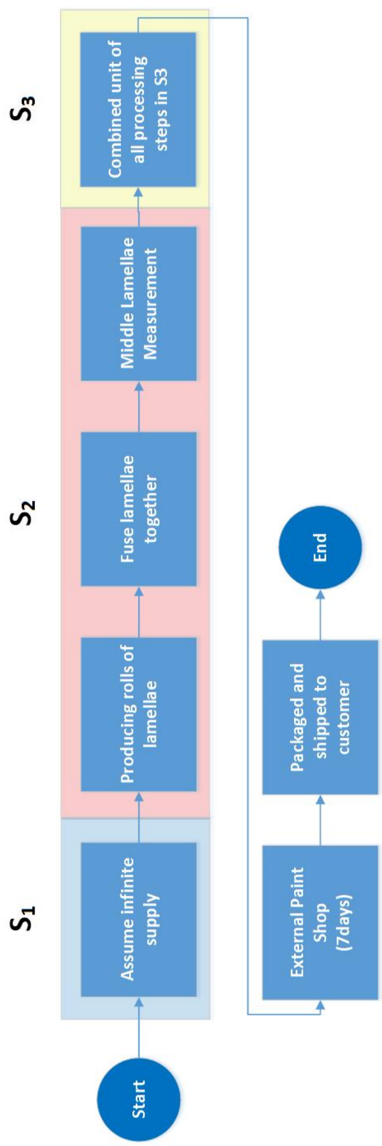


Figure 44: Process Flow of a regular grid after combining stages

C Input Parameters Stage 2

Processing Times

Table 13: Processing Time (min), Operation 1 (Matten)

	M1	M2	M3
P_t	15	15	15

Table 14: Processing Time (sec), Operation 2 (Plakken) for machines using the STB method

	M11	M12	M13	M14	M15	M16	M17	M18	M20	M21	M22	M23	M24	M25	M26	M28	M29	M30	M31	M32	M33	M34	M35	M36
P_t	4.8	4.97	5.208	5.508	5.447	5.6	4.9	5.5	5.2	5.2	4.85	5	5.7	5	5.227	5.9	5	5.84	5.2	4.847	5.7	5.4	6	5.4

Table 15: Processing Time (sec), Operation 2 (Plakken) for machines using the LTB method

	M50	M51	M52	M53	M54	M55	M56	M57	M58	M59
P_t	7.29	7.21	7.11	7.11	7.56	7.52	7.52	7.08	7.34	7.41

Table 16: Processing Time (min), Operation 3 (MLM)

	M88
P_t	3

Setup Times

Table 17: Setup Time (min), Operation 1 (Matten) for all machines

	Height difference	Line type difference	No difference
S_t	60	10	5

Table 18: Setup Time (sec), Operation 2 (Plakken) for all machines which use the STB method

	Height difference	Line type difference	No difference
S_t	55	18	15

Table 19: Setup Time (sec), Operation 2 (Plakken) for all machines which use the LTB method

	Always
S_t	15

Table 20: Setup Time (min), Operation 3 (MLM)

	Always
S_t	1

Eligible Operators

Table 21: Machines where operators are eligible to perform setups

OMM	1,2,3
OPP	11,12,13,14,15,16,17,18,20,21,22,23,24,25,26,28,29,30,31,32,33,34,35,36,50,51,52,53,54,55,56,57,58,59
OML	88

Demand per day

Table 22: Average demand for 1 day, part 1

Job	Height	Line type	Number of Lamellae	Method
J0	1.5	70L	2415	STB
J1	1.5	70L	2415	STB
J2	1.5	70L	3470	STB
J3	1.5	70L	2657	STB
J4	1.5	70L	2657	STB
J5	1.5	70L	2657	STB
J6	2.0	44L	1205	STB
J7	1.5	41L	1622	STB
J8	1.5	41L	1622	STB
J9	1.5	41L	1622	STB
J10	1.5	41L	1622	STB
J11	1.5	41L	1622	STB
J12	1.5	31L	1173	STB
J13	1.5	31L	1173	STB
J14	1.5	31L	1173	STB
J15	1.5	60L	1610	STB
J16	1.5	60L	1610	STB
J17	2.5	44L	1563	LTB
J18	2.5	44L	1563	LTB
J19	2.5	44L	1563	LTB
J20	2.5	44L	1563	LTB
J21	2.5	44L	1563	LTB
J22	2.5	44L	1563	LTB
J23	2.5	44L	1563	LTB
J24	2.5	44L	1563	LTB
J25	2.5	44L	1563	LTB
J26	2.0	44L	2178	STB
J27	2.0	44L	2178	STB
J28	1.5	60L	2277	STB
J29	3.0	44L	2178	STB
J30	3.0	44L	2178	STB
J31	3.0	44L	2178	STB
J32	1.5	41L	1504	STB
J33	1.5	41L	1504	STB
J34	1.5	41L	1504	STB
J35	1.5	41L	1504	STB
J36	1.5	41L	1504	STB

Table 23: Average demand for 1 day, part 2

Job	Height	Line type	Number of Lamellae	Method
J37	1.5	41L	1504	STB
J38	1.5	41L	1504	STB
J39	1.5	41L	1504	STB
J40	1.5	41L	1504	STB
J41	1.5	41L	1504	STB
J42	1.5	41L	1504	STB
J43	1.5	41L	1504	STB
J44	1.5	74L	1983	STB
J45	1.5	74L	1983	STB
J46	1.5	74L	1983	STB
J47	1.5	74L	1983	STB
J48	1.5	74L	1983	STB
J49	1.5	74L	1983	STB
J50	1.5	80L	3220	STB
J51	1.5	80L	3220	STB
J52	1.5	67L	2200	LTB
J53	1.5	67L	2200	LTB
J54	1.5	67L	2200	LTB
J55	1.5	67L	2200	LTB
J56	1.5	67L	2200	LTB
J57	1.5	67L	2200	LTB
J58	1.5	67L	2200	LTB
J59	1.5	67L	2200	LTB
J60	1.5	67L	2200	LTB
J61	1.5	67L	2200	LTB
J62	1.5	67L	2200	LTB
J63	2.5	44L	2352	LTB
J64	2.5	44L	2352	LTB
J65	1.75	40L	1980	LTB
J66	1.75	40L	1980	LTB
J67	1.5	70L	1852	STB
J68	2.5	36L	1782	STB
J69	2.5	36L	1782	STB
J70	2.5	36L	1782	STB
J71	1.5	60L	1546	STB
J72	1.5	60L	1546	STB
J73	1.5	60L	1546	STB
J74	1.5	60L	1546	STB
J75	1.5	60L	1546	STB
J76	1.5	60L	1546	STB
J77	1.5	60L	1546	STB
J78	1.5	60L	1546	STB
J79	1.5	60L	1546	STB
J80	2.0	40L	2121	STB
J81	1.75	40L	2112	STB
J82	1.75	40L	2112	STB
J83	1.75	40L	2112	STB
J84	1.75	40L	2112	STB
J85	1.75	40L	2112	STB

Table 24: Average demand for 1 day, part 3

Job	Height	Line type	Number of Lamellae	Method
J85	1.75	40L	2112	STB
J86	1.75	40L	2112	STB
J87	1.75	40L	2112	STB
J88	1.75	40L	2112	STB
J89	1.75	40L	2112	STB
J90	1.75	40L	2112	STB
J91	1.5	70L	2343	STB
J92	1.5	70L	2343	STB
J93	1.5	70L	2343	STB
J94	1.5	60L	2125	STB
J95	1.5	74L	2000	STB
J96	1.5	74L	2000	STB
J97	1.5	74L	2000	STB
J98	1.5	74L	2000	STB
J99	2.0	44L	2159	LTB
J100	1.5	60L	2093	STB
J101	1.5	60L	2093	STB
J102	1.5	60L	2093	STB
J103	1.5	60L	2093	STB
J104	1.5	60L	2093	STB
J105	1.5	70L	1827	STB
J106	1.5	44L	1834	LTB
J107	1.5	44L	1834	LTB
J108	1.75	60L	2125	STB
J109	2.0	60L	2346	STB
J110	1.5	74L	2834	STB
J111	1.5	74L	2834	STB
J112	1.5	60L	2346	STB
J113	1.5	60L	2346	STB
J114	1.5	60L	2346	STB
J115	1.5	60L	2346	STB
J116	1.5	60L	2346	STB
J117	1.5	60L	2346	STB
J118	1.5	60L	1656	STB
J119	1.5	60L	1656	STB
J120	2.5	40L	2112	STB
J121	2.5	40L	2112	STB
J122	2.5	40L	2112	STB
J123	2.5	40L	2112	STB
J124	2.5	40L	2112	STB
J125	1.5	70L	2632	LTB
J126	1.5	70L	2608	LTB
J127	1.5	74L	2042	STB

D Reorder Levels

Table 25: Reorder levels, 31L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	10	12	12		12	14	15		21	21	23
avg D	12	**12**	12		14	15	17		24	24	24
high D	12	12	13		15	16	17		25	26	28

Table 26: Reorder levels, 36L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	6	6	7		6	8	9		11	11	12
avg D	7	**8**	9		8	8	10		12	14	14
high D	8	10	11		9	10	11		15	16	17

Table 27: Reorder levels, 40L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	11	13	13		16	16	17		34	36	37
avg D	11	**11**	11		16	18	17		36	40	43
high D	13	13	13		18	18	19		47	47	48

Table 28: Reorder levels, 41L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	8	9	10		13	13	13		34	34	34
avg D	10	**11**	11		16	17	17		43	43	43
high D	11	11	12		17	17	18		44	44	45

Table 29: Reorder levels, 44L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	14	15	16		19	20	21		32	35	37
avg D	15	**17**	16		20	21	20		38	39	40
high D	19	20	20		23	24	24		48	50	51

Table 30: Reorder levels, 50L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	2	2	3		2	2	3		3	3	3
avg D	2	**3**	3		2	3	3		3	3	3
high D	2	3	4		2	3	4		3	4	4

Table 31: Reorder levels, 52L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	4	5	6		5	6	6		8	8	9
avg D	5	**5**	7		6	7	8		8	10	12
high D	6	7	8		7	8	9		11	13	13

Table 32: Reorder levels, 57L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	5	6	6		5	6	7		6	7	8
avg D	6	**7**	7		7	7	7		7	8	8
high D	6	7	8		6	7	8		8	9	10

Table 33: Reorder levels, 60L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	24	26	27		30	30	31		56	57	59
avg D	26	**28**	29		32	33	33		64	67	70
high D	26	28	29		32	34	33		68	70	73

Table 34: Reorder levels, 67L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	7	8	10		8	9	11		11	12	13
avg D	9	**10**	12		10	11	14		12	14	17
high D	10	11	14		12	13	16		14	16	19

Table 35: Reorder levels, 70L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	24	24	24		28	28	29		49	49	50
avg D	25	**25**	26		30	30	32		53	53	56
high D	26	26	26		31	31	32		58	58	62

Table 36: Reorder levels, 74L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	19	22	26		21	25	28		28	30	35
avg D	25	**28**	26		24	29	31		33	37	38
high D	26	29	30		28	30	32		40	41	42

Table 37: Reorder levels, 80L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	10	12	14		11	12	15		15	18	20
avg D	11	**13**	14		12	15	16		18	22	23
high D	14	15	15		15	15	16		23	24	24

Table 38: Reorder levels, 85L R=1, R=2, R=7

	low LT	avg LT	high LT		low LT	avg LT	high LT		low LT	avg LT	high LT
low D	6	10	11		8	10	11		11	12	13
avg D	9	**10**	12		9	11	13		12	13	15
high D	10	12	13		11	12	15		14	16	18

E Hyper-parameters GA

Table 39: Fitness values when k=2

$P_m \backslash P_c$	0.8	0.85	0.9	0.95
0.05	405821	419764	427382	427994
0.1	422638	427424	434758	435992
0.15	429635	430110	442841	440090
0.2	444031	439452	452286	445456

Table 40: Fitness values when k=3

$P_m \backslash P_c$	0.8	0.85	0.9	0.95
0.05	391767	376388	368161	362765
0.1	377876	361323	358361	359295
0.15	376290	365662	356521	363887
0.2	370763	364795	361665	401479

Table 41: Fitness values when k=4

$P_m \backslash P_c$	0.8	0.85	0.9	0.95
0.05	398066	406993	369492	373145
0.1	392267	390845	363558	361003
0.15	388397	375426	360325	366280
0.2	375830	375096	364732	370775

F Operator Comparison Stage 2

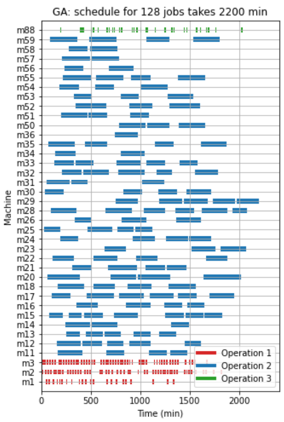


Figure 45: 2 Operators for Operation 2

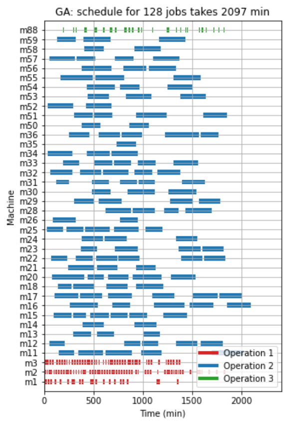


Figure 46: 3 Operators for Operation 2

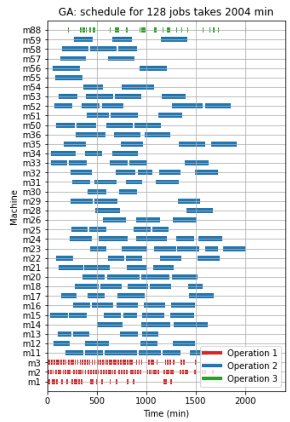


Figure 47: 4 Operators for Operation 2

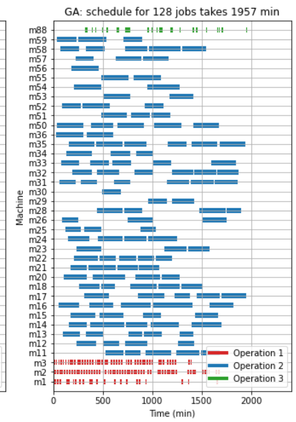


Figure 48: 5 Operators for Operation 2

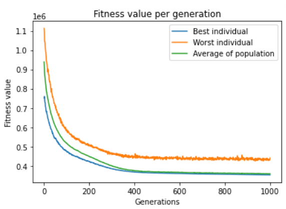


Figure 49: Convergence Plot, 2 Operators for Operation 2

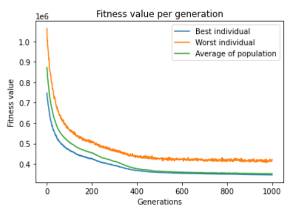


Figure 50: Convergence Plot, 3 Operators for Operation 2

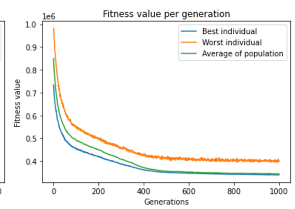


Figure 51: Convergence Plot, 4 Operators for Operation 2

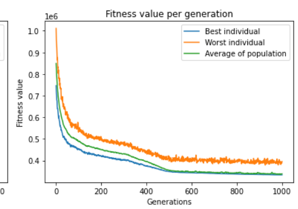


Figure 52: Convergence Plot, 5 Operators for Operation 2

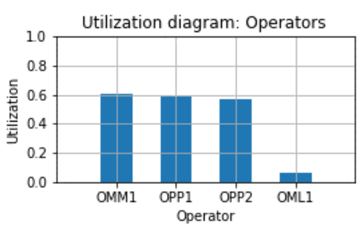


Figure 53: 2 Operators for Operation 2

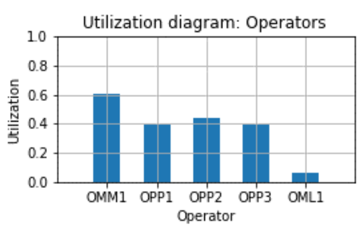


Figure 54: 3 Operators for Operation 2

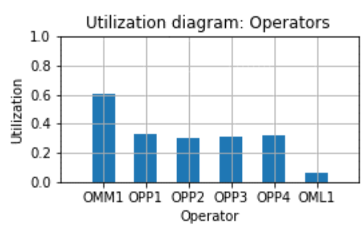


Figure 55: 4 Operators for Operation 2

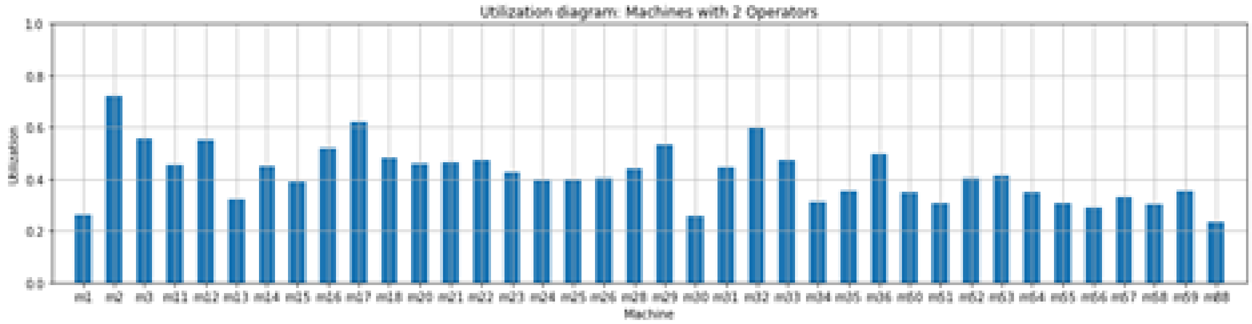


Figure 56: 2 Operators for Operation 2

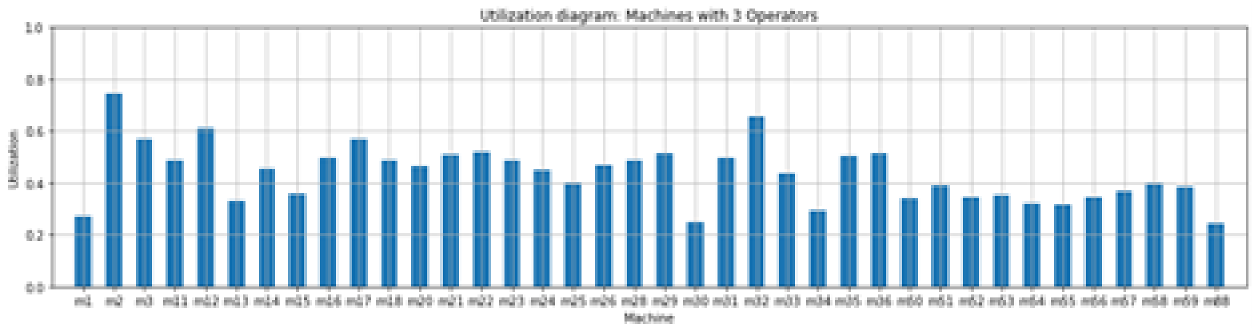


Figure 57: 3 Operators for Operation 2

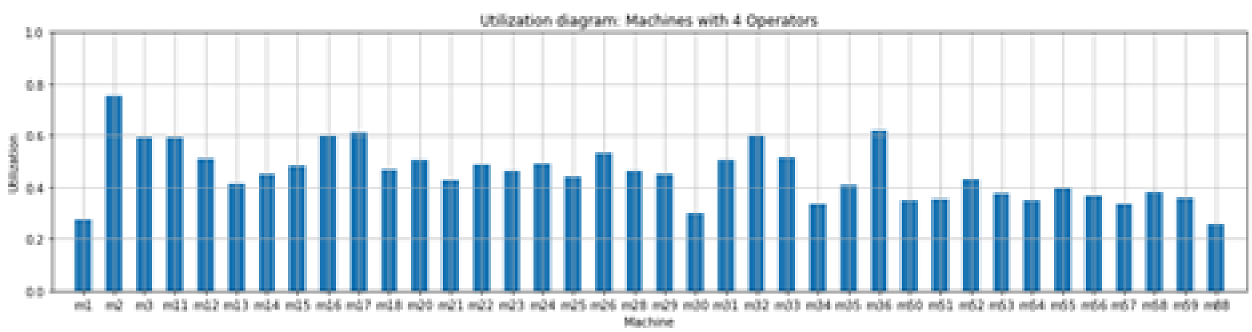


Figure 58: 4 Operators for Operation 2

G Grid Type Analysis Stage 2

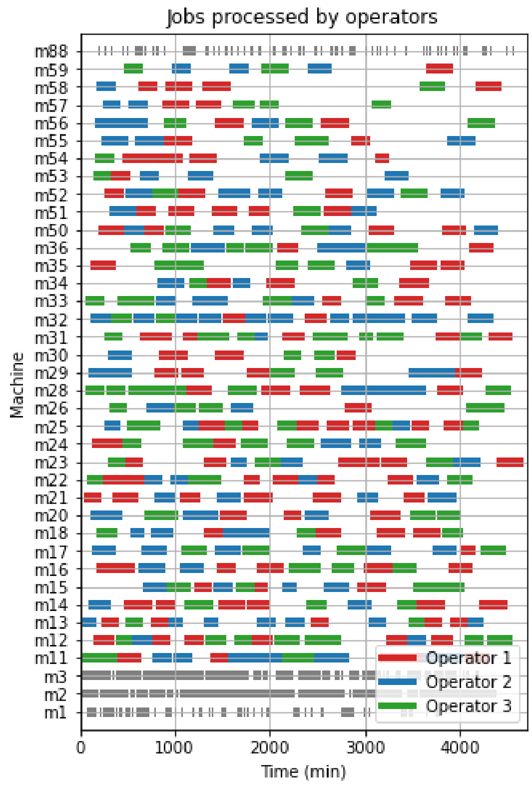


Figure 59: Setup performed by operator

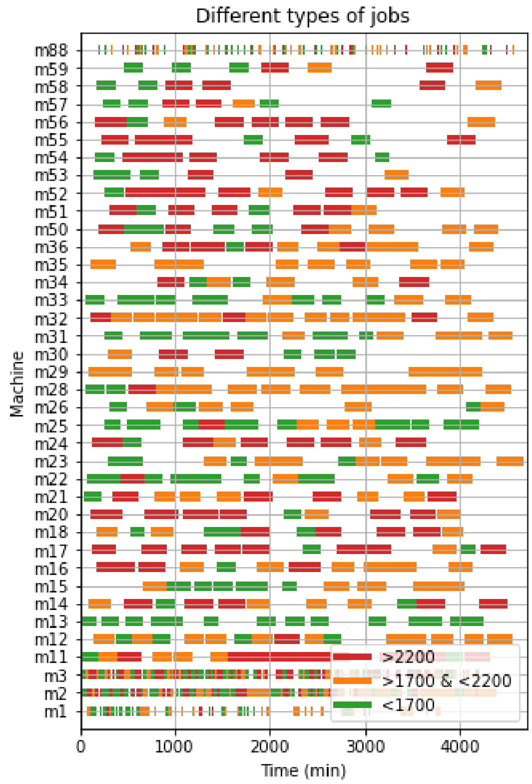


Figure 60: Jobs categorized based on number of lamellae

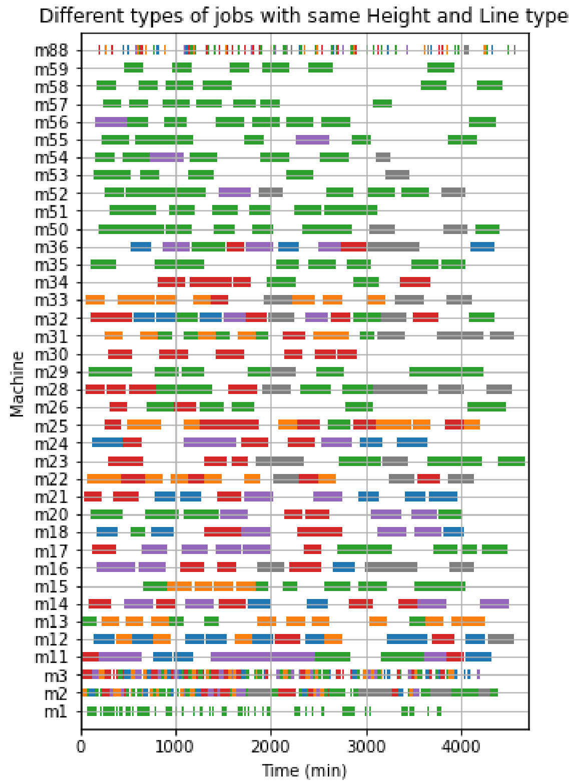


Figure 61: Jobs categorized based on height and line type

H Activities Stage 3

Post-processing 1	Veneer	Quality	Post-processing 2	Distribution
Washing grids	Plastic veneer	Quality check1	Scratch off grids	Add extras
Sawing	Veneer	Quality check	Milling	pack up
Cleaning grids	Rolling grids	MLM	Add extras	Sending
Unpack carbon	Vacuum		Mill framework	storage control
Plane carbon	Stamping		produce different framework	clean hallway
Cut carbon	Plane carbon			Order chemicals
Creating round grids	Cut carbon			unpack holders
Washing frameworks	Creating round grids			
	Extra cleaning			
	Scratch off mammography grids			

I Bottleneck Stage 3

Post-processing 1	Veneer	Post-processing 2	P-p 1	Veneer	P-p 2	Bottleneck rate
1-1-1	1-1-1	2-2-1	110	115	195	110
1-1-1	1-1-1	2-2-0	110	115	130	110
1-1-1	1-1-1	1-1-1	110	115	120	110
1-1-1	1-1-1	1-1-0	110	115	85	85
1-1-1	2-1-1	2-2-1	110	150	195	110
1-1-1	2-1-1	2-2-0	110	150	130	110
1-1-1	2-1-1	1-1-1	110	150	120	110
1-1-1	2-1-1	1-1-0	110	150	85	85
1-1-1	1-1-0	2-2-1	110	80	195	80
1-1-1	1-1-0	2-2-0	110	80	130	80
1-1-1	1-1-0	1-1-1	110	80	120	80
1-1-1	1-1-0	1-1-0	110	80	85	80
2-2-2	1-1-1	2-2-1	150	115	195	115
2-2-2	1-1-1	2-2-0	150	115	130	115
2-2-2	1-1-1	1-1-1	150	115	120	115
2-2-2	1-1-1	1-1-0	150	115	85	85
2-2-2	2-1-1	2-2-1	150	150	195	150
2-2-2	2-1-1	2-2-0	150	150	130	130
2-2-2	2-1-1	1-1-1	150	150	120	120
2-2-2	2-1-1	1-1-0	150	150	85	85
2-2-2	1-1-0	2-2-1	150	80	195	80
2-2-2	1-1-0	2-2-0	150	80	130	80
2-2-2	1-1-0	1-1-1	150	80	120	80
2-2-2	1-1-0	1-1-0	150	80	85	80
2-2-1	1-1-1	2-2-1	125	115	195	115
2-2-1	1-1-1	2-2-0	125	115	130	115
2-2-1	1-1-1	1-1-1	125	115	120	115
2-2-1	1-1-1	1-1-0	125	115	85	85
2-2-1	2-1-1	2-2-1	125	150	195	125
2-2-1	2-1-1	2-2-0	125	150	130	125
2-2-1	2-1-1	1-1-1	125	150	120	120
2-2-1	2-1-1	1-1-0	125	150	85	85
2-2-1	1-1-0	2-2-1	125	80	195	80
2-2-1	1-1-0	2-2-0	125	80	130	80
2-2-1	1-1-0	1-1-1	125	80	120	80
2-2-1	1-1-0	1-1-0	125	80	85	80

Post-processing 1	Veneer	Post-processing 2	P-p 1	Veneer	P-p 2	Bottleneck rate
2-1-1	1-1-1	2-2-1	100	115	195	100
2-1-1	1-1-1	2-2-0	100	115	130	100
2-1-1	1-1-1	1-1-1	100	115	120	100
2-1-1	1-1-1	1-1-0	100	115	85	85
2-1-1	2-1-1	2-2-1	100	150	195	100
2-1-1	2-1-1	2-2-0	100	150	130	100
2-1-1	2-1-1	1-1-1	100	150	120	100
2-1-1	2-1-1	1-1-0	100	150	85	85
2-1-1	1-1-0	2-2-1	100	80	195	80
2-1-1	1-1-0	2-2-0	100	80	130	80
2-1-1	1-1-0	1-1-1	100	80	120	80
2-1-1	1-1-0	1-1-0	100	80	85	80
1-1-0	1-1-1	2-2-1	65	115	195	65
1-1-0	1-1-1	2-2-0	65	115	130	65
1-1-0	1-1-1	1-1-1	65	115	120	65
1-1-0	1-1-1	1-1-0	65	115	85	65
1-1-0	2-1-1	2-2-1	65	150	195	65
1-1-0	2-1-1	2-2-0	65	150	130	65
1-1-0	2-1-1	1-1-1	65	150	120	65
1-1-0	2-1-1	1-1-0	65	150	85	65
1-1-0	1-1-0	2-2-1	65	80	195	65
1-1-0	1-1-0	2-2-0	65	80	130	65
1-1-0	1-1-0	1-1-1	65	80	120	65
1-1-0	1-1-0	1-1-0	65	80	85	65
2-2-0	1-1-1	2-2-1	100	115	195	100
2-2-0	1-1-1	2-2-0	100	115	130	100
2-2-0	1-1-1	1-1-1	100	115	120	100
2-2-0	1-1-1	1-1-0	100	115	85	85
2-2-0	2-1-1	2-2-1	100	150	195	100
2-2-0	2-1-1	2-2-0	100	150	130	100
2-2-0	2-1-1	1-1-1	100	150	120	100
2-2-0	2-1-1	1-1-0	100	150	85	85
2-2-0	1-1-0	2-2-1	100	80	195	80
2-2-0	1-1-0	2-2-0	100	80	130	80
2-2-0	1-1-0	1-1-1	100	80	120	80
2-2-0	1-1-0	1-1-0	100	80	85	80
2-1-0	1-1-1	2-2-1	65	115	195	65
2-1-0	1-1-1	2-2-0	65	115	130	65
2-1-0	1-1-1	1-1-1	65	115	120	65
2-1-0	1-1-1	1-1-0	65	115	85	65
2-1-0	2-1-1	2-2-1	65	150	195	65
2-1-0	2-1-1	2-2-0	65	150	130	65
2-1-0	2-1-1	1-1-1	65	150	120	65
2-1-0	2-1-1	1-1-0	65	150	85	65
2-1-0	1-1-0	2-2-1	65	80	195	65
2-1-0	1-1-0	2-2-0	65	80	130	65
2-1-0	1-1-0	1-1-1	65	80	120	65
2-1-0	1-1-0	1-1-0	65	80	85	65