Improving the On-Time Delivery Performance at PM

MASTER'S THESIS INDUSTRIAL ENGINEERING & MANAGEMENT

Author: L. Antonides

February – July 2019

UNIVERSITY OF TWENTE.

Master's Thesis Industrial Engineering & Management

Specialization: Production & Logistics Management

Author

Lydia Antonides

University

University of Twente
Drienerlolaan 5
7522 NB Enschede

Thesis conducted at

PM B.V.
Galileistraat 2
7701 SK Dedemsvaart

Supervisors University of Twente

Dr. Ir. J.M.J. Schutten
Dr. Ir. A.G. Maan - Leeftink

Supervisors PM

G. Lennips – Operations Manager
D. Horenberg – Assembly Manager

Date

23 July 2019

Management Summary

Steadily growing over the years, Precisie Metaal (PM) now encounters the challenge of maintaining a high On-Time Delivery Performance (OTDP). PM is a family-run company located in Dedemsvaart, which manufactures amongst others high-precision linear and rotating bearings. Despite its excellent product quality, PM experiences dissatisfied customers due to long lead times and deliveries that arrive too late. Its current OTDP is 80.3%. To make sure that the current customers keep ordering at PM, PM wants to achieve an OTDP of 95%.

During preliminary research we discovered that the scheduling process does not accurately represent reality, which leads to schedules that are not realizable and thus due dates are not met, which leads to a reduced OTDP. Reality is not accurately represented because workers are not included as a constraint in the scheduling process, while they are. Also, the currently used scheduling tool at PM, Factory Planning, is not optimally utilized. This tool offers the ability to apply set-up optimization. This can reduce the required machine capacity.

The goal of this research is to gain knowledge on what the opportunities are in optimizing the scheduling process to create more realistic and improved schedules that will increase the OTDP at PM. To achieve this goal the main question of this research is:

"How can PM improve its production schedules to increase the on-time delivery performance?"

To answer this research question we first analyze the current situation. We define Key Performance Indicators (KPIs) to assess the performance of the current scheduling process. Second, we review the literature available on this topic to evaluate the proposed scheduling solutions in literature. Then we look at what extensions Factory Planning offers to improve the production schedules. Finally, we evaluate various scheduling algorithms and run experiments to assess and compare the performance in terms of the defined KPIs.

Relevant performance indicators to assess the performance of the currently created production schedules are the On Time Delivery Performance (OTDP), Average Number of Days Late (ADL), Average Set-up Times (AST) and Average Queueing Time (AQT). The current performance of PM is based on data of 2018. The ADL of the orders shipped is 17.6 days. The AST is 8.4% of the total processing time. The current AQT at the production departments of PM is 7.6 days.

To make the schedules represent reality more accurately, we consider worker capacity as an additional constraint next to machine capacity which is currently the only resource constraint that PM considers in their scheduling process. Considering this dual resource constrained scheduling problem in a job shop environment, we review the literature on the Dual Resource Constraint Flexible Job shop Scheduling Problem (DRCFJSP). We observe that the Genetic Algorithm (GA) is a very common approach in literature for various scheduling problems. To improve the local search ability of the GA, we are interested in the hybrid metaheuristic GA-VNS. An interesting addition is the robustness of the schedule against unforeseen disturbances on the shop floor. This can be achieved by minimizing the lateness.

We investigate two approaches to solve the scheduling problem. The first approach considers extending the current planning system of PM, Factory Planning. Including set-up optimization, which aims to reduce the sequence dependent set-up time by matching succeeding items based on their

product characteristics, helps to reduce the set-up time. Reduced set-up times result in reduced lead times that contributes to achieving an improved OTDP. The importance of implementing set-up optimization is not just to reduce the set-up and thus the lead time, but also to represent reality more accurately, because Factory Planning does not detect coincidental set-up time reduction when set-up optimization is not included. This inaccuracy of Factory Planning could cause disturbances on the shop floor and lead to reduced efficiency.

Because the experimenting possibilities with Factory Planning are rather limited and to investigate the option of a new scheduling algorithm, we address a second approach. We review several scheduling algorithms, the Genetic Algorithm – Variable Neighborhood Search (GA-VNS), Genetic Algorithm – Neighborhood Search (GA-NS), Simulated Annealing (SA), Steepest Decent (SD), and both the GA-VNS and GA-NS enhanced by an additional SD, respectively GA-VNS+SD and GA-NS+SD. We consider these algorithms in 4 scheduling scenarios.

Table S1 shows the results of the experiments we execute to compare the various algorithms for the 4 scenarios. The values in this table are the lateness in days, which we aim to minimize. The green values indicate the best performance for each scenario. Table S1 presents the lateness values without considering the actual worker constraint and the coincidal set-up optimization. Table S2 shows the results of the same experiments, but these results are corrected for the worker constraint and the set-up optimization by coincidence that both occur in practice. So this second table more closely resembles reality. The values printed in italics indicate that the results are the same as for Table S1, because the experiment does already include both the worker constraint and set-up optimization in the optimization process of the schedules.

Algorithm	**Scenario 1: No SU, No Worker**	**Scenario 2: SU, No Worker**	**Scenario 3a: No SU, Worker**	**Scenario 3b: SU, Worker**	**Scenario 4a: No SU, Semi-auto**	**Scenario 4b: SU, Semi-auto**
GA-VNS	-194.7	-194.0	-112.5	-113.5	-114.9	-118.4
GA-NS	-215.2	-211.3	-139.7	-124.9	-136.7	-132.6
SA	**-232.7**	**-228.5**	**-153.4**	-118.6	**-157.0**	-129.7
SD	21.6	-39.9	152.2	147.8	150.7	141.9
GA-VNS + SD	-197.6	-196.3	-116.1	-116.5	-118.5	-121.9
GA-NS + SD	-216.9	-212.4	-141.6	**-128.0**	-138.2	**-136.3**

Table S1

Algorithm	**Scenario 1: No SU, No Worker**	**Scenario 2: SU, No Worker**	**Scenario 3a: No SU, Worker**	**Scenario 3b: SU, Worker**	**Scenario 4a: No SU, Semi-auto**	**Scenario 4b: SU, Semi-auto**
GA-VNS	0.7	48.4	-112.5	*-113.5*	-114.9	*-118.4*
GA-NS	-16.6	30.8	-139.7	*-124.9*	-136.7	*-132.6*
SA	**-28.2**	**22.3**	**-153.5**	*-118.6*	**-157.0**	*-129.7*
SD	181.3	183.9	152.2	*147.8*	150.7	*141.9*
GA-VNS + SD	0.8	47.2	-116.1	*-116.5*	-118.5	*-121.9*
GA-NS + SD	-16.8	30.6	-141.6	***-128.0***	-138.2	***-136.3***

Table S2

Including set-up optimization does reduce the lateness. The effect of actively optimizing the set-up is however limited because it sometimes occurs that set-up time can be neglected because by chance two sequential items have the same set-up parameter values. Including set-up optimization does however reduce the AST. This does not directly reduce the OTDP, but is does reduce the lead time.

Reduced lead times make it possible to deliver faster which is an important aspect of customer service next to OTDP. Including the worker constraint reduces the lateness both when set-up optimization is included and when this is not included. We also see that each scheduling algorithm is capable of improving the schedule quality in terms of lateness when this constraint is considered in the scheduling algorithm. Considering semi-automatic machines slightly reduces the lateness. These results are however not statistically significant. By evaluating the performance of each scheduling algorithm on each of the scenarios, we observe that both SA and GA-NS+SD perform best. In case we include all scheduling extensions, set-up optimization, worker constraint and semi-automatic machines, the GA-NS+SD performs best.

Because PM's current scheduling tool is compatible with their ERP system, Glovia, and it offers a lot of possibilities to enhance the quality of the schedules we think for now it is undesirable to invest time and money in a new scheduling algorithm. To improve the production schedules in order to increase the on-time delivery performance, we recommend PM on the short term to:

- Implement set-up optimization in their current scheduling process.
- Implement the worker constraint in Factory Planning.
- Consider include semi-automatic machines in the scheduling process.
- Critically reassess the current data in Glovia.

On the long term, we recommend PM to:

- Conduct additional research on the performance of SA and GA-NS + SD compared to Factory Planning.
- Conduct additional research on the possibilities of connecting a new scheduling algorithm to Glovia.

Depending on the results of this additional research, PM can, after improving the performance of Factory Planning, improve the performance of the schedules even more by considering an alternative scheduling algorithm that also includes the scheduling extensions that we recommend to implement for Factory Planning and also applies a smarter scheduling logic that is custom-made for PM.

Preface

The past months I spent working on my master's thesis. I really enjoyed working on this real-life case at PM where I learned a lot. Working on my master's thesis was interesting, sometimes difficult and there were moments I thought I would never reach the end of my thesis. I want to thank my family and friends who supported me and helped me to get through the difficult moments.

I would like to thank Marco Schutten and Gréanne Maan-Leeftink for supervising me during my master's thesis. I appreciate the feedback they provided me with and the time they took to discuss with me how to proceed during my research. Also I would like to thank Gert Lennips and Dominic Horenberg for providing me with this opportunity to execute my graduation project at PM. They provided me with the tools and opportunities within the company to be able to execute my research. I admire and appreciate their openness for new ideas and innovations. For my research I also needed information and support from several employees at PM. I would like to thank these colleagues for patiently helping me out.

Now that I am about to receive my master's degree in Industrial Engineering and Management I am closing off the period of being a student. I had a good time the past five years I was a student at the University of Twente. I started off with my bachelors in Health Sciences, but the enthusiasm of Erwin Hans, professor at the University of Twente, made me make the right decision of proceeding with my masters in Industrial Engineering and Management. I had the opportunity to follow interesting and challenging courses which helped me to develop tools and skills and made me excited for my future career. I think I have a lot more to learn and I cannot wait to start developing myself as a professional.

Lydia Antonides,

Zwolle, July 2019

Table of Contents

List of Abbreviations

ADL	Average Days Late
AQT	Average Queueing Time
AST	Average Set-up Time
BOM	Bill Of Materials
DRCFJSP	Dual Resource Constrained Flexible Job-shop Scheduling Problem
ERP	Enterprise Resource Planning
GA	Genetic Algorithm
JSP	Job-shop Scheduling Problem
KPI	Key Performance Indicators
NS	Neighbourhood Search
OTDP	On-Time delivery Performance
SA	Simulated Annealing
SD	Steepest Descent
VNS	Variable Neighbourhood Search

1 Introduction

Precisie Metaal (PM) experiences a too low delivery performance. This is a crucial element of the customer service, because being able to deliver quickly and on time is an opportunity to stand out among competitors. This chapter describes the problem that PM experiences and the relevance of solving this problem. Section 1.1 briefly introduces PM. Section 1.2 elaborates on the problem that is encountered. Section 1.3 describes the research questions and the approach to answer these questions.

1.1 Company Introduction

PM is a family-run company, located in Dedemsvaart, which manufactures high-precision linear and rotating bearings. Figure 1 and Figure 2 show two examples of products developed and produced by PM (PM, Innovation, 2019a). PM also develops custom-made, high quality systems that are applied in, amongst others, the semiconductor industry, factory automation and medical sciences environments. PM uses a make to stock (MTS), make to order (MTO) and engineer to order (ETO) production approach. PM was founded in 1966 and still is an independent business. Over the years PM has grown to a company of over 200 employees (PM, 2019b).

Figure 1: Rotating Bearing

Figure 2: Gonio Bearing

1.2 Problem Description

Over the past years, PM has been growing steadily. Figure 3 shows the annual revenue for PM over the past years, including an expected revenue for the current year, 2019. This growth introduced the challenge of maintaining a high on-time delivery performance.

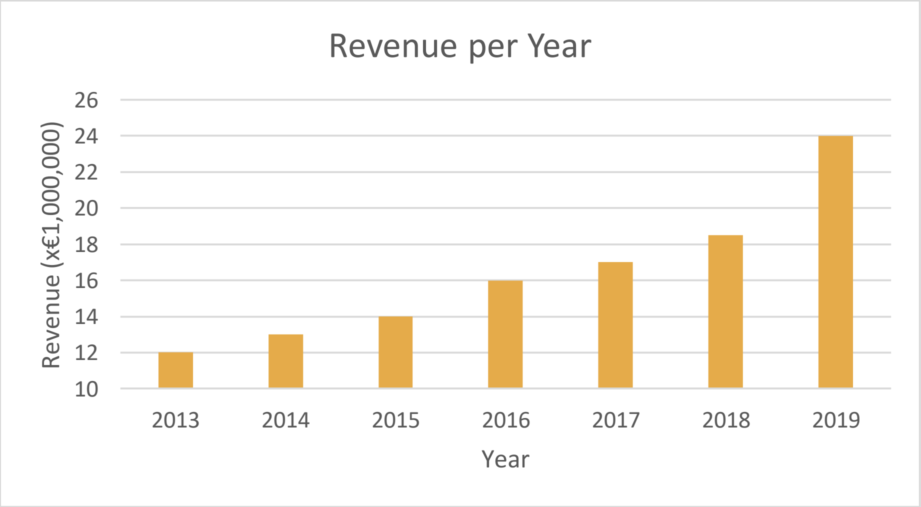


Figure 3: Revenue per Year

Despite its excellent product quality, PM experiences dissatisfied customers due to long lead times and deliveries that arrive too late. Its current on-time delivery performance is 80.3%. To make sure that the current customers keep ordering at PM, PM wants to improve the on-time delivery performance. The aim is to achieve an on-time delivery performance of at least 95%. Although an on-time delivery performance higher than 95% is not necessarily desirable, because this is likely to be at the expense of another performance indicator. For example, if the machine utilization is low enough, which can be achieved by accepting less sales orders, an on-time delivery performance of 100% can be achieved. This is of course not desirable because the revenue will be lower due to less accepted sales orders.

PM distinguishes between production and assembly departments. Figure 4 depicts the average queueing time of the production departments and of the assembly departments. We observe that products mainly have to wait during the production process. Waiting is often unnecessary and extends the lead time. The necessary queueing times, for example time that is needed for the products to cool down, is excluded. This figure shows that the problems especially occur at the production departments.

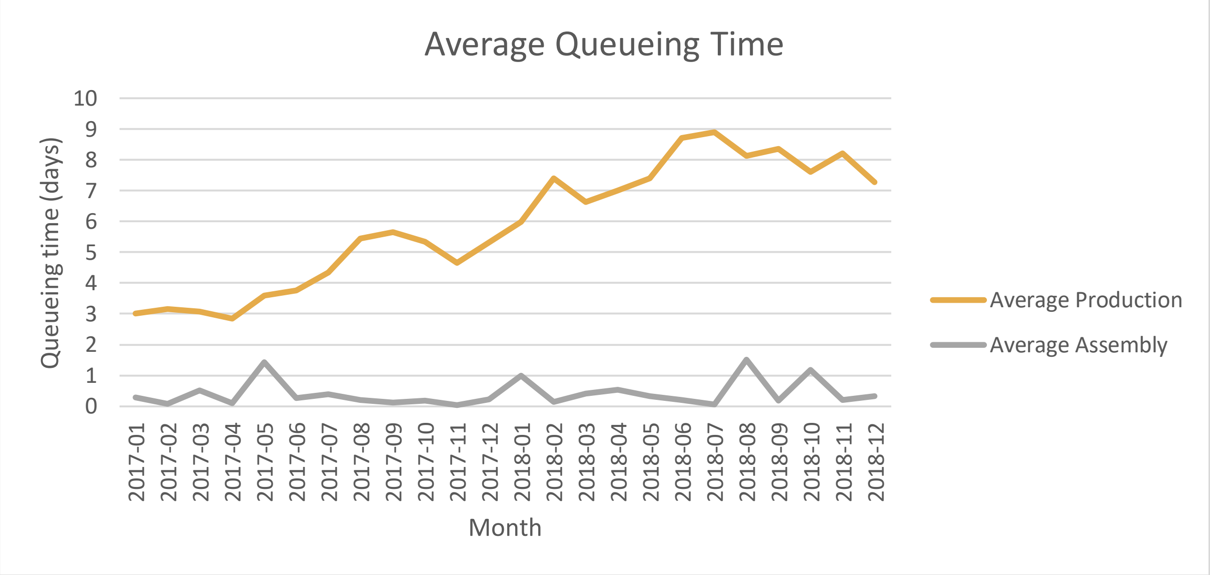


Figure 4: Queueing Time Production and Assembly

Figure 5 depicts the relationship between the observed problem of an insufficient on-time delivery performance and its underlying causes. Orders are not delivered according to the date agreed with the customer, because often the production due date is not met. Capacity at most production departments is not enough to process all the orders according to the schedule. The fact that capacity is not sufficient has four causes. First of all, machines break down every now and then. Currently, PM does not attempt to prevent this. This is Core Problem 1. Second, the total set-up time is high. This reduces the effective production time available at the machines. The total set-up time is high, because when the orders are scheduled, the opportunity to minimize the set-up times is not considered. This is Core Problem 2. Third, the number of workers to operate the machines is not always sufficient. This is caused by the fact that this constraint is not included in the scheduling process, Core Problem 3, and by the fact that workers cannot be flexibly deployed, because of required training and experience to execute certain tasks. The latter is Core Problem 4. Fourth, there are too many orders scheduled. This is caused by orders that are accepted on a short term. Since only part of the orders is forecasted, demand cannot fully be known in advance, so not all required capacity can be scheduled on a long term. This is Core Problem 5.

We identify five core problems in this problem cluster. We do not consider Core Problem 1 for research for several reasons. For part of the failures it makes no sense to estimate when they will occur, because most failures do not have a great impact. When a minor failure occurs, it can immediately be solved by, in most cases, replacement of tooling, and production can continue. As for the large impact failures, preventive maintenance in the form of an external maintenance contract turned out to not be worthwhile based on past experience at PM.

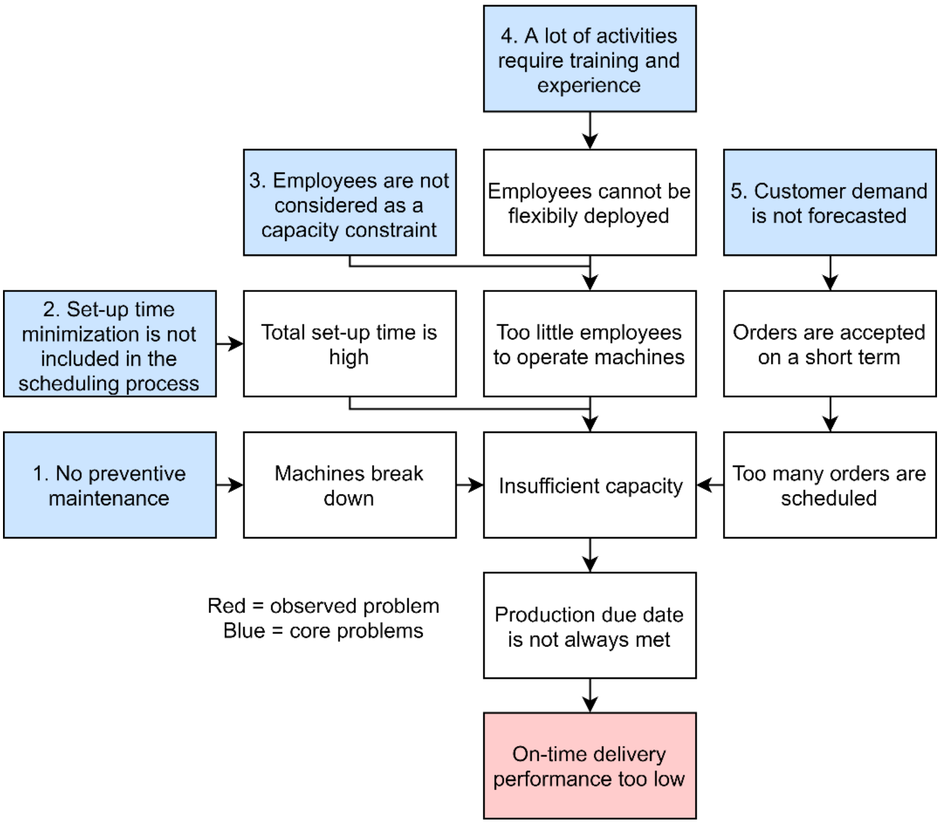

4. A lot of activities require training and experience
3. Employees are not considered as a capacity constraint
Employees cannot be flexibily deployed
5. Customer demand is not forecasted
2. Set-up time minimization is not included in the scheduling process
Total set-up time is high
Too little employees to operate machines
Orders are accepted on a short term
1. No preventive maintenance
Machines break down
Insufficient capacity
Too many orders are scheduled
Red = observed problem
Blue = core problems
Production due date is not always met
On-time delivery performance too low

Figure 5: Problem Cluster

Core Problem 2 has great improvement opportunities. According to the production manager, time savings can be realized if the orders are scheduled such that the set-up times are minimized for example by clustering orders that require the same machine settings. Also, Core Problem 3 is an aspect that shows room for improvement. In the scheduling process only the machine capacities are taken into account. However, often a worker is required to start up a machine or needs to attend the machine during the complete processing time. So, if the constraint of available workers is considered in the scheduling process, the resulting schedule will more realistically resemble the actual situation, which will increase the probability that the order can be finished by the due date discussed with the customer. Core Problem 4 is hard to influence. The activities encountered in the production process require a certain level of training and experience, i.e. skill. To adjust this level of required skill, adjustments to the technical specifications of the products need to be made. This concerns the engineering part of the processes, which is out of scope of this research. It is also a possibility to train workers to achieve the required skill level. This, however, comes at a cost. There is a trade-off between worker skill level and the investment cost to attain this skill level. The last core problem, Core Problem 5 would be interesting to look at, because being able to predict the demand makes it easier to plan ahead. However PM is partly an ETO and MTO production environment. This makes the future customer demand very stochastic, which makes it hard to forecast what resources will be required when customers request a newly engineered product. Addressing this core problem is more suitable for an environment in which all products are high volume, low variety products. To conclude, in this research we address Core Problems 2 and 3. So, the main focus of the research is the optimization of the generated schedules.

1.3 Research Approach

The goal of this research is to gain knowledge on what the opportunities are in optimizing the scheduling process to create more realistic and improved schedules which will increase the on-time delivery performance at PM. To achieve this goal the main question of this research is:

"How can PM improve its production schedules to increase the on-time delivery performance?"

PM currently uses a planning system called Factory Planning to generate the production schedules. To improve the schedules at PM we consider on the one hand available extensions that Factory Planning offers and on the other hand we develop a new scheduling heuristic. We consider a new scheduling heuristic, because we think that Factory Planning cannot offer all required scheduling features PM requires. For the scheduling process, we mainly focus on the MTO part of the production processes at PM, because this is the most complex to schedule. The ETO process is similar to the MTO process, except for the engineering step. For this research we are interested in the processes at the production departments. Because the engineering step takes place before the processing steps at the production departments starts, we can exclude this step from this research.

To support the process of finding an answer to the main research question, we formulate several sub questions:

Sub Question 1: *What is the performance of the currently generated schedules?*
Chapter 2 addresses Sub Question 1. This chapter describes the current scheduling and production processes, discusses various Key Performance Indicators (KPIs) and the current performance. Finally, this chapter reviews additional scheduling constraints.

Sub Question 2: *What methods for achieving an optimized production schedule in an MTO environment are described in literature?*
Chapter 3 consists of the literature review and answers Sub Question 2. This chapter first discusses MTO environments in general, followed by an extensive review of the literature on job shop scheduling processes and relevant extensions of this problem.

Sub Question 3: *What are potential scheduling improvements for PM?*
Sub Question 3a: *How can existing extensions of the planning system of PM be implemented?*
Chapter 4 answers Sub Question 3a. This chapter addresses the relevant extensions of Factory Planning. This chapter also describes how we apply these extensions. Finally Chapter 4 presents the results of applying these extensions to Factory Planning.
Sub Question 3b: *What should a new and optimal scheduling process for PM look like?*
Chapter 5 addresses Sub Question 3b. This chapter first presents the problem at hand and the solution approach. Then this chapter explains the computational experiment procedure. Subsequently Chapter 5 presents the results of these experiments.

Sub Question 4: *What is the impact of the scheduling improvements for PM?*
Chapters 4 and 5 both include a section on the results of the proposed scheduling improvements for respectively Factory Planning and the new scheduling process.

Chapter 6 concludes this research and presents the conclusions based on the answers on each of the sub questions. This chapter also includes advice on the scheduling process in relation to the on-time delivery performance at PM and therewith answers our main research question.

Plan of Approach

We execute the following steps to answer **Sub Question 1**, *"What is the performance of the currently generated schedules?"*:

1. Analyze the current production and scheduling process.
2. Assess the performance of the current schedules.
3. Evaluate what additional scheduling constraints are valuable for PM to consider.

We execute the following steps to answer **Sub Question 2**, *"What methods for achieving an optimized production schedule in an MTO environment are described in literature?"*:

1. Execute literature research.
2. Rate the heuristics and methods found in literature on their applicability for PM.

We execute the following steps to answer **Sub Question 3**, *"What are potential scheduling improvements for PM?"*:

1. Determine the parameters that are required to implement the existing extensions of the planning system of PM.
    a. Talk to developers of PM's scheduling system.
    b. Study the manuals of PM's scheduling system.
    c. Implement parameters experimentally in PM's scheduling system.
2. Develop a scheduling heuristic.
    a. Determine all characteristics of the scheduling problem at hand.
    b. Integrate the relevant scheduling heuristics found at Sub Question 2 for this scheduling problem.
    c. Implement scheduling heuristic in Visual Studio.

We execute the following steps to answer **Sub Question 4**, *"What is the impact of the scheduling improvements for PM?"*:

1. Assess the impact of the extended existing planning system.
    a. Experiment in PM's scheduling system.
    b. Calculate the achieved performance based on the Key Performance Indicators.
2. Assess the impact of the new scheduling algorithm.
    a. Run experiments with the implemented scheduling heuristic.
    b. Calculate the achieved performance based on the Key Performance Indicators.
3. Analyze what solution is most promising for PM.
    a. Draw conclusions based on the results found.
    b. Give advice on implementation at PM.
    c. Give advice on future research.

Scope of Research

This research only considers the production process steps. The assembly process steps are not considered, because the observed problems that cause the on-time delivery to be too low occur at the production departments. The production process steps precede the assembly processing steps. This research only considers PM in Dedemsvaart and its customers. Other parts of the supply chain are excluded, because it makes sense to first optimize the in-house processes, before considering external factors, which might be harder to influence.

2 Context Analysis

To answer the first research question: *"What is the performance of the currently generated schedules?"* we conduct a context analysis. Section 2.1 describes the current production and scheduling process. Section 2.2 discusses the relevant Key Performance Indicators (KPIs). Section 2.3 reviews what scheduling extensions are valuable for PM. Section 2.4 concludes this chapter.

2.1 Current Production and Scheduling Process

PM uses both an ETO and MTO production approach. Part of the products is produced MTS, this encompasses the so called standard products. The layout at PM is a process layout, also called a job shop or jobbing environment, see Figure 6. The various machines are clustered by their function. In this way separate departments can be identified. Each department has its own function. At PM there are five separate production departments, namely the cutting, drilling, milling, hardening and grinding departments. Such a production environment typically has a low production volume and a high product variety. This is also partly the case for PM. The ETO products are specifically designed for a certain customer, so the variety of products is high, and the quantity is low. This is mainly because part of the products ordered are not standard products, so other customers will not order the same products with the same specifications. As for MTO, products are based on a standard design, but the exact final product is based on customer's specifications (Krajewski, Malhotra, & Ritzman, 2016). Variation in the product specification of various customers makes it inconvenient to keep a lot of items in stock.

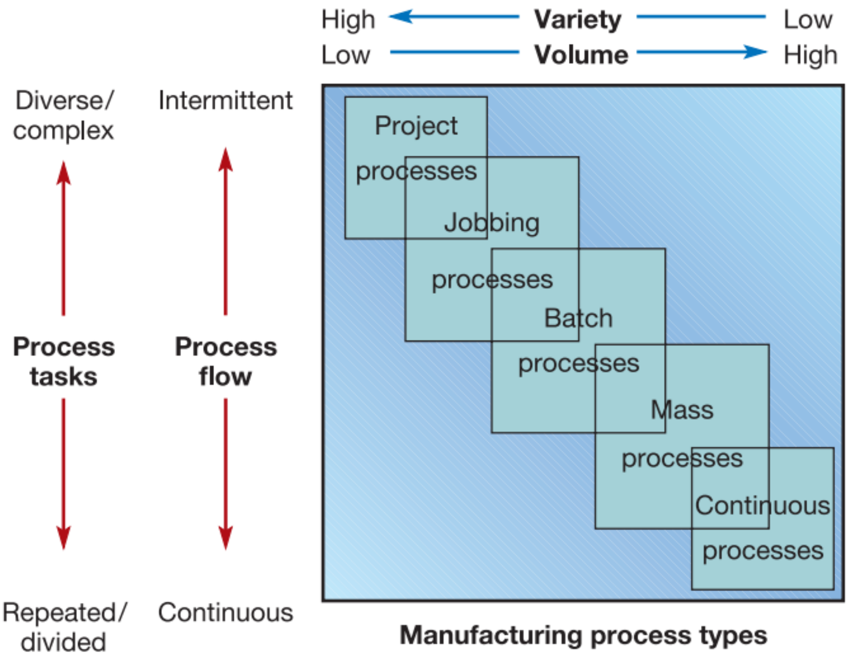


Figure 6: Volume - Variety Matrix (Slack, Brandon-Jones, & Johnston, 2013)

The products produced according to an ETO and MTO approach, are scheduled for production based on a customer order. Once scheduled, the products are pushed through the system. This is defined as a push strategy, because the order is triggered by an external customer. The MTS products are scheduled for production as soon as the inventory has decreased to a certain level. This internal trigger makes this a pull strategy. In the remainder of this section, we first discuss the production process, followed by the initiation process of an ETO and MTO order. Next, we discuss the MTS process. Finally we address the scheduling process.

Production Process

By "job" we denote the complete production process of a product. A job consists of several sequential operations, which are specific for each product. The sequence of operations is called the routing. To give an illustration of the production process without going into too much detail of each separate routing, we discuss the main routing steps, which roughly have a logical sequence. The actual routing of each product, however, deviates from the routing that we discuss here. For example, some products require a step multiple times, other products skip certain steps, or two steps are swapped. Figure 7 depicts these main routing steps. All orders require raw material to start with. The right raw material is selected at the cutting department. The raw material is cut at the right length according to the product specifications. From the cutting department the products are transported to either the drilling department or the milling department, depending on the product specifications. Some processing steps can be done either at the drilling or the milling department. Within the drilling department also a lathe machine and a precision electrical chemical machining (PEM) machine are located. After the required drilling and milling processing steps, the products need to be hardened. All parts are hardened in house, except for stainless steel parts. The hardening of these parts is outsourced to Pontus. Sequentially, products that have a linear shape, such as linear bearings, need to be straightened. This is done by hand, and each product is handled individually. The first straightening steps are done at the hardening department. If further straightening is required, then this will be done at the grinding department. This is also the final department the products go to in the production process. Next to additional straightening, products undergo several grinding operations to acquire their accuracy. When all surfaces of a certain part are grinded, this part is measured in the measuring department, which is also located at the grinding department, to check whether its accuracy is sufficient. If the measurements show insufficient accuracy, then this part might require additional grinding processing or when this is not possible, this part is rejected. When the manufacturing is finished, the parts are stored until they are further processed by the assembly department.

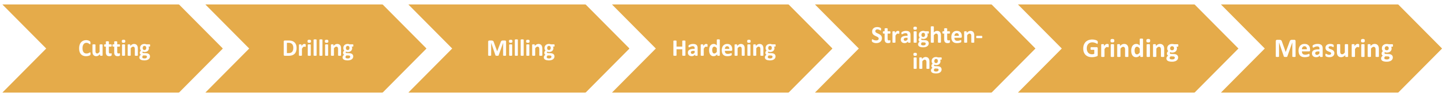


Figure 7: Production Process

ETO and MTO

The ETO process is similar to the MTO process with exception of the engineering step. The engineering step precedes all the other processing steps, which are also present for the MTO process. For simplification we will only consider this MTO process. Figure 8 depicts these process steps. The MTO production process is initiated by a customer order. We assume that at this point the engineering process is finished. This customer order requests a certain quantity of a certain product or multiple products. The sales department translates the request of the customer to a sales order in the enterprise resource planning (ERP) system Glovia. In some cases, the customer has a requested date. This is the date by when the customer would like to receive the products. In most cases the customer wants the products as soon as possible. If an order has a requested date, the planner will check whether it is possible to finish the products of the order before the requested date. He does this by performing a Capable To Promise (CTP) check in the planning system, Factory Planning. This check computes a CTP date by adding the new order to the already scheduled orders and recalculating a

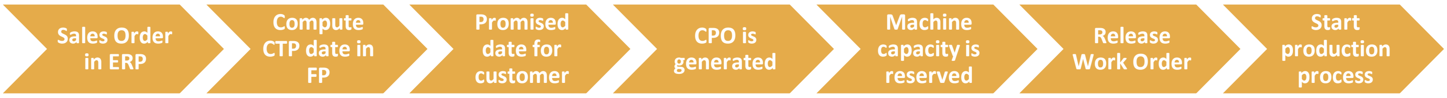


Figure 8: Work Order Planning Process

schedule including this new order. This date is computed such that the order will be finished before the requested date of the customer. In the second case, when customers want to receive the order as soon as possible, the current date will be entered as the requested date. In this way Factory Planning will schedule this new order such that it is finished as soon as possible. The CTP check results in a 'promised date'. This is the date in terms of weeks, that is communicated to the customer. The internal 'scheduled date' is about one week before the promised date. This week constitutes scheduled safety time. It can occur that the CTP date is later than the requested date of the customer. In this case, the planner manually needs to find a way to be able to finish this order earlier. This can be done by producing the product on a different machine than initially was determined, by outsourcing or by producing in overtime. When the CTP date does not cause any problems, this order can be scheduled, and a Computer Planned Order (CPO) is made. Once an order is present as a CPO, capacity of the relevant machines is reserved. At this point a work order, which is required to initiate the production process, is not yet released. A work order is released when two conditions are met. First of all, an order can only be released if all required materials are present. If one or multiple parts or materials are not available, the order will not be released. Second, the order will be released only if the start date advised by Factory Planning is at most one week in the future. Orders that have a start date further in the future will not yet be released to prevent an overload of active work orders and too high Work In Process (WIP). As soon as a work order is released, the production of the products can start. The work order is delivered to the first production step, which is in most cases the cutting department. From there on the work order and the (partially finished) products travel through the production process.

MTS

Part of the products produced at PM is MTS. The number of products produced MTS is preferably small, because PM wants to reduce the risk of losing money due to products in stock. When producing MTS, money might be lost due to products that remain unsold. Also, the money that is invested in these products cannot be used for other purposes. For some products a certain level of safety stock is maintained. This is done for standard products that are often requested by customers. Standard products are produced in batches to reduce production costs. For these products an order is released as soon as the stock drops to a certain level. The actual stock levels are monitored through the ERP system Glovia. When a new batch of a certain product is required, a CPO for this product is added into the planning system. Products are also made to stock when a customer orders less than the production batch size. The remaining products that the customer did not order but were produced because of the batch size, are stored until they are requested by a customer order.

Factory Planning

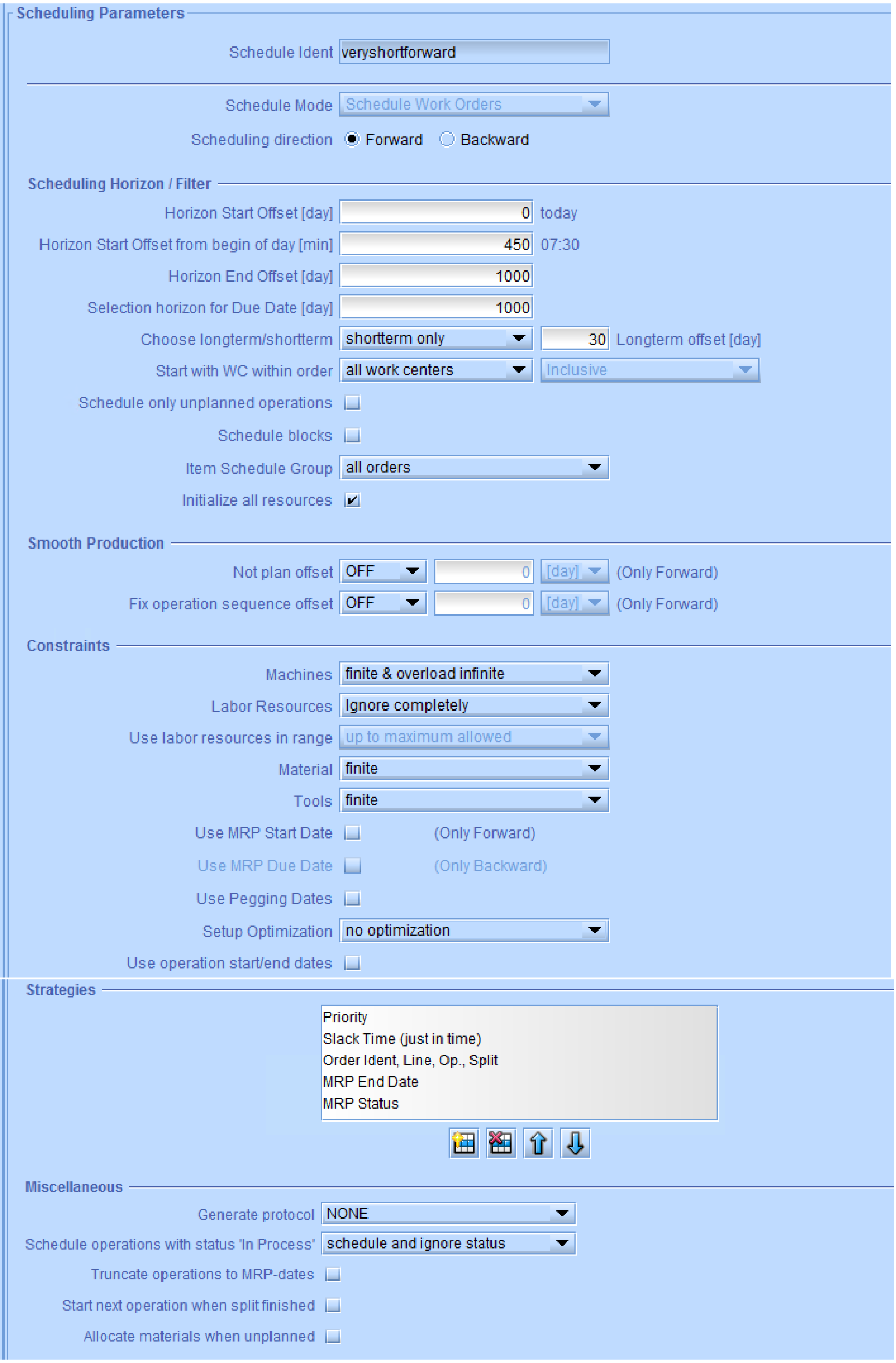
Factory Planning is the planning system that PM uses since 2016 to schedule their orders. This system is connected to the ERP system, Glovia. To determine a due date for the customer, Factory Planning uses the CTP check. This process generates a due date taking into account the resource capacity and the expected lead time. For the CTP check, Factory Planning first creates the so-called CTP orders based on the customer request, the sales order. For this sales order, one or multiple CTP work- and purchase orders are created, depending on the lower-level product requirements. The bill of materials (BOM) expands to multiple levels and the required lower-level CTP work- and purchase orders are created. Based on the required date, the CTP check calculates the start date. This calculation is based on the expected lead times of the product and the machine capacity. This start date now serves as a required date for the lower level orders. In this way, the CTP check calculates all start dates until the lowest component level is reached. Due dates for purchase orders are calculated as the current date plus the lead time. This is the First CTP Test. A scheduling run in the CTP check can be unsuccessful if the calculated start date is in the past. When this First CTP Test is unsuccessful, the Alternative CTP Test is run. To explain what these First and Alternative CTP Test do, we first explain what scheduling parameters can be set in Factory Planning.

The user of Factory Planning can create multiple scheduling idents. A scheduling ident is a set of certain scheduling parameters. This scheduling ident results in a certain scheduling process depending on the values of the various parameters. Factory Planning can run multiple scheduling idents sequentially. Many parameters can be set in Factory Planning, of which we discuss five. The first parameter is the scheduling direction. The scheduling direction can either be forward or backward. For forward scheduling, Factory Planning starts scheduling from the current date or a user-defined date in the future towards the end of the defined scheduling horizon. For backward scheduling, Factory Planning starts scheduling from a user-defined date in the future, backwards to the current date, or another date between the current date and the end of the scheduling horizon if defined by the user. This also addresses the second parameter: the scheduling horizon. Only orders that fall in the horizon defined by the user are scheduled. By setting this parameter, the user can let Factory Planning execute long term or short-term scheduling. The third parameter is the machine setting. The machine setting can either be finite or infinite. If set to finite, Factory Planning only schedules an order if the capacity of the relevant machine is sufficient. If the required machine does not have enough capacity and the scheduling direction is forward, the order is scheduled further into the future, as soon as the capacity is sufficient. It is not clear what Factory Planning does if the scheduling direction is backward and the machine capacity is set to finite. Although this combination is possible in Factory Planning, the manual does not address this option. If set to infinite, Factory Planning does not consider how much capacity of the machine still is available, it only considers the requested due date and expected lead times. The fourth parameter is the materials setting. This setting can also be set as either finite or infinite. If set to finite, material is a constraint. If infinite, the available materials are not taken into account. The fifth parameter concerns the resource initialization. By setting this parameter, the user can define whether Factory Planning on a new scheduling run should reset all resources, in this case the machines, and schedule all orders again including the new to be scheduled orders, or should schedule the new to be scheduled orders as an addition to the already existing schedule.

The First CTP test contains the following scheduling parameter settings: backwards scheduling, planning horizon is 365 days, machines are finite, material is infinite, machines are not initialized. The Alternative CTP Test contains the following scheduling strategy settings: forward scheduling, planning horizon is 365 days, resources are finite, material is finite, resources are not initialized.

Once the CTP check is successful, this procedure results in a start date and a due date. This scheduled due date plus a few days for secondary operations is the promised date that PM communicates to the customer. At this point the order is scheduled as a CPO and the required machine capacity is reserved. About one week prior to the calculated start date the scheduled order is released to the work floor. Before an order can be released, the CPO is converted to a work order.

Next to the CTP Test scheduling idents, the planner can define a schedule sequence to reschedule the orders. Currently three scheduling idents are defined in Factory Planning which are run sequentially when the planner wants an updated schedule. The three scheduling idents are ‘Veryshortforward’, ‘Shortforward’ and ‘Longbackward’. Figure 9 shows how the scheduling parameters per scheduling ident can be set in Factory Planning. Appendix A: Factory Planning Details describes the details of the current scheduling settings in Factory Planning. Veryshortforward only schedules the orders that are due within 30 days into the future. This first scheduling ident schedules in a forward direction and starts off by initializing all



Scheduling Parameters

Schedule Ident: veryshortforward

Schedule Mode: Schedule Work Orders

Scheduling direction: ● Forward ○ Backward

Scheduling Horizon / Filter

Horizon Start Offset [day]: 0 today

Horizon Start Offset from begin of day [min]: 450 07:30

Horizon End Offset [day]: 1000

Selection horizon for Due Date [day]: 1000

Choose longterm/shortterm: shortterm only 30 Longterm offset [day]

Start with WC within order: all work centers Inclusive

Schedule only unplanned operations □

Schedule blocks □

Item Schedule Group: all orders

Initialize all resources ☑

Smooth Production

Not plan offset: OFF 0 [day] (Only Forward)

Fix operation sequence offset: OFF 0 [day] (Only Forward)

Constraints

Machines: finite & overload infinite

Labor Resources: Ignore completely

Use labor resources in range: up to maximum allowed

Material: finite

Tools: finite

Use MRP Start Date □ (Only Forward)

Use MRP Due Date □ (Only Backward)

Use Pegging Dates □

Setup Optimization: no optimization

Use operation start/end dates □

Strategies

Priority
Slack Time (just in time)
Order Ident, Line, Op., Split
MRP End Date
MRP Status

Miscellaneous

Generate protocol: NONE

Schedule operations with status 'In Process': schedule and ignore status

Truncate operations to MRP-dates □

Start next operation when split finished □

Allocate materials when unplanned □

Figure 9: Current Settings Veryshortforward Factory Planning

machines, so at the start the schedule is empty. Shortforward, the second scheduling ident, schedules all orders of the coming year, but does not schedule these orders on the first two next days. The scheduling direction is forward, and the results of the previous scheduling ident are not reset, resources are not initialized. The final scheduling ident, longbackward, fills up the schedule in a backward fashion and starts 1000 days into the future. Since this is a long-term schedule run, the machine capacities are not considered as a constraint. Again, previous scheduling results are not neglected.

Each of these three scheduling idents applies a scheduling strategy that consist of certain priority rules that are applied in hierarchical order to determine what job to schedule next. For each of the three scheduling idents the scheduling strategy used is the same. The scheduling strategy is as follows. First the job with highest priority is scheduled. The priority can be set manually in Factory Planning. If multiple jobs have the same priority, the job with the least slack time is scheduled. If multiple jobs have the same amount of slack time, the job with the highest order-number is scheduled next. Since this number is unique for each job, any additional scheduling strategy will not be used.

Factory Planning Extensions

Factory Planning contains a range of possibilities to extend the scheduling process. The user can use different scheduling strategies, define scheduling groups, encounter additional constraining resources, and set-up optimization. Factory Planning consists of many more options to fine tune the scheduling process and add detail. Here we only discuss the main extensions that we expect to impact the scheduling performance.

The current scheduling strategy consists of five priority rules. Next to priority, slack time, order-number, MRP end date and MRP status, there are a few additional priority rules that the user can select: First In First Out (FIFO), Longest Operation Time, Shortest Operation Time and In-Process First. The user can define what rules to use in what hierarchical order.

To add more detail to the scheduling process, the user can define scheduling groups. These groups contain different types of orders. For example short-term MTO items or pre-built items. The user can then define for each group what its scheduling priority is, i.e. which group should be scheduled first, which second, and so on. The user can also define for each group what scheduling strategy should be applied. These extensions can be useful to assign a priority value to a group of jobs instead to jobs individually. Also, this adds a hierarchical scheduling strategy level, because within a scheduling group, which has a certain priority, each job can also have a priority value.

Instead of machines, Factory Planning can also encounter labor and tools as a capacity resource. This will restrict the schedule possibilities because a resource constraint is added, but on the other hand including labor as a capacity constraint makes the schedule more realistic, which makes it more likely that the schedule is realizable. When the labor resources are enabled as a scheduling constraint in addition to machine resource constraint, Factory Planning also checks whether there are sufficient workers to schedule a certain job on a certain machine at a certain time. A labor resource is an individual person, with a certain number of hours available per workday. This person can have one or multiple skills. A skill is the capability of a worker to attend a certain machine. When multiple labor resources are available with the same skill then a priority is assigned to each worker. A worker with least skills, gets highest priority. This worker will be scheduled first as soon as it has sufficient skills.

Factory Planning contains a set-up optimization option. When enabled, Factory Planning, determines what order should be scheduled next to minimize the required set-up time. Before set-up optimization can be applied, the user must determine what parameters influence the set-up time. Then, for each product the value for each parameter needs to be defined in Glovia. For example if the color of a product determines the set-up time then "color" is the parameter. The value of this parameter can for example be white. In Factory Planning the set-up matrix need to be filled in, which gives information on what the consequences are of switching from one parameter value to another. This switching occurs when the values of a parameter of two succeeding products is different. The required set-up time depends on the value of both the preceding and succeeding product. For example, switching from a black to white color might require more set-up time than switching from white to black if cleaning is required when switched to lighter colors. The scheduling process can handle up to five set-up optimization parameters. Factory Planning deals with them in hierarchical order. In the scheduling process, Factory Planning calculates what order should be scheduled next in order to minimize the set-up time.

PM does currently not apply all these extensions that Factory Planning offers. Workforce scheduling and set-up optimization are the scheduling improvement opportunities that are interesting for PM to improve the production schedules generated by Factory Planning. Apart from the extensions that this chapter discusses, not all extensions are relevant for PM. Factory Planning is a tool developed for a broad range of industries and is not customized for PM. So, it is important to critically assess what extensions to invest time in and what parameters for the extensions are relevant to experiment with to get its right values.

2.2 Key Performance Indicators

The goal of this research is to improve the on-time delivery performance. This is an important factor for the customer service level. Another factor that influences the customer service level and that is also related to order delivery, is short lead times. We first discuss the on-time delivery. Next we discuss the short lead times.

Whether orders can be delivered on time, depends on how realizable the schedule is. We assume that the scheduled date at which the order is planned to be finished according to Factory Planning, is such that the delivery date arranged with the customer, is possible to meet according to the schedule. If the schedule is, however, not realizable, for example because it does not represent the actual situation on the shop floor, it is likely the order will not be delivered on time. To measure how realizable a schedule is, we define two KPIs, namely the percentage of orders delivered on time and the average number of days orders arrive too late if an order is too late.

Lead times can be reduced if the workload is efficiently scheduled. By efficient scheduled we mean a schedule that effectively utilizes the available resources in order to produce as efficiently as possible. In an ideal situation orders are scheduled such that the time that makes up the lead time only consists of actually processing the order. In this situation the product does not have to wait, which means that the lead time is minimized. Completely excluding queueing time is however unrealistic, but this illustrates that reducing queueing time will reduce the lead time. To measure the schedule efficiency, we define two KPIs, namely the average set-up times and the average queueing time of orders, because time spent on setting up and waiting could be removed from the total lead time, to achieve shorter lead times. Figure 10 displays the relationship of the relevant KPIs. To get the baseline measurements we use data of 2017 and 2018.

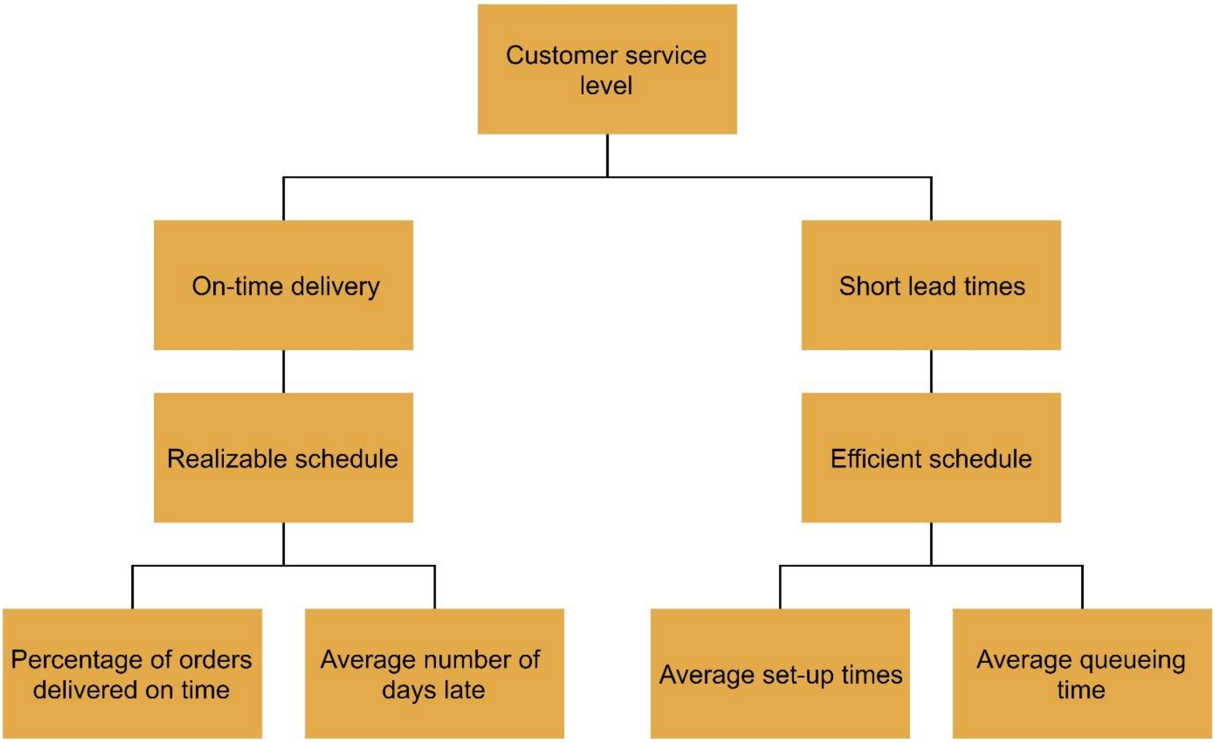


Figure 10: KPIs Relationship Diagram

Percentage of Orders Delivered on Time

An order is considered on time if the shipped date is in the same week as the promised date. So, a late order is a sales order that is shipped any day after the week of the promised date. The promised date is only used internally as a guideline. The date promised to the customer, however is in terms of weeks. The week of the promised date, complies with the week that is communicated to the customer as promised delivery date. In the remainder we refer to this KPI as the on-time delivery performance (OTDP) and is calculated as follows:

$$On - time\ delivery\ performance = \left(1 - \frac{Number\ of\ late\ orders}{Total\ number\ of\ order\ shipments}\right) * 100\%$$

The OTDP can be calculated for different time intervals. The data included for a certain time interval are based on the shipped date. So, if the shipped date of an order lies within the time interval that is being considered, this order is included.

Figure 11 depicts the weekly OTDP for 2017 and 2018. This figure also depicts the OTDP of the year 2018. The OTDP of 2018 is 80.3%. This value is the baseline measurement of the OTDP.

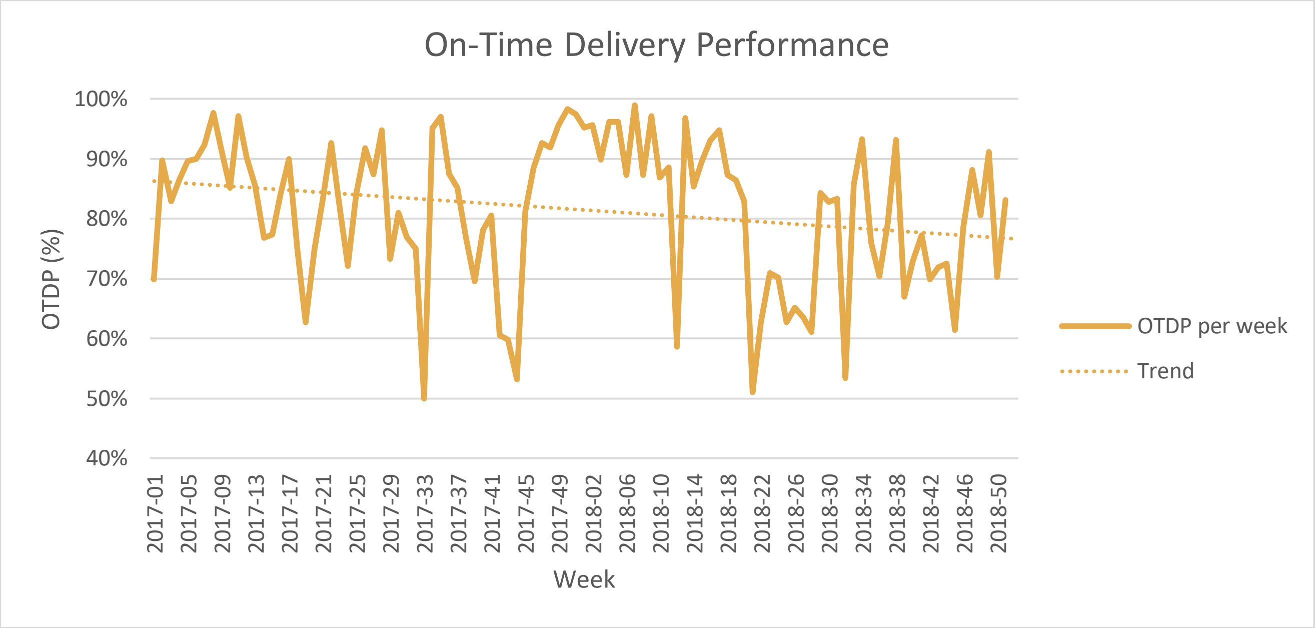


Figure 11: On-Time Delivery Performance

Average Number of Days Late

For the average number of days late (ADL) we only consider the orders that were too late. This KPI is calculated using the following formula:

$$Average\ number\ of\ days\ late = \frac{Total\ number\ of\ days\ late\ of\ late\ orders}{Number\ of\ late\ orders}$$

The ADL can be calculated for different time intervals. The data included for a certain time interval are based on the shipped date. So, if the shipped date of an order lies within the time interval that is being considered, this order is included.

Figure 12 depicts the weekly ADL for 2017 and 2018. The ADL calculated over 2018 is 17.6. This value is the baseline measurement of the ADL. Due to two extreme values, which are caused by a few orders that have a very high number of days late, the graph is not very detailed. To get a clearer view of the graph, Figure 13 shows a close up.

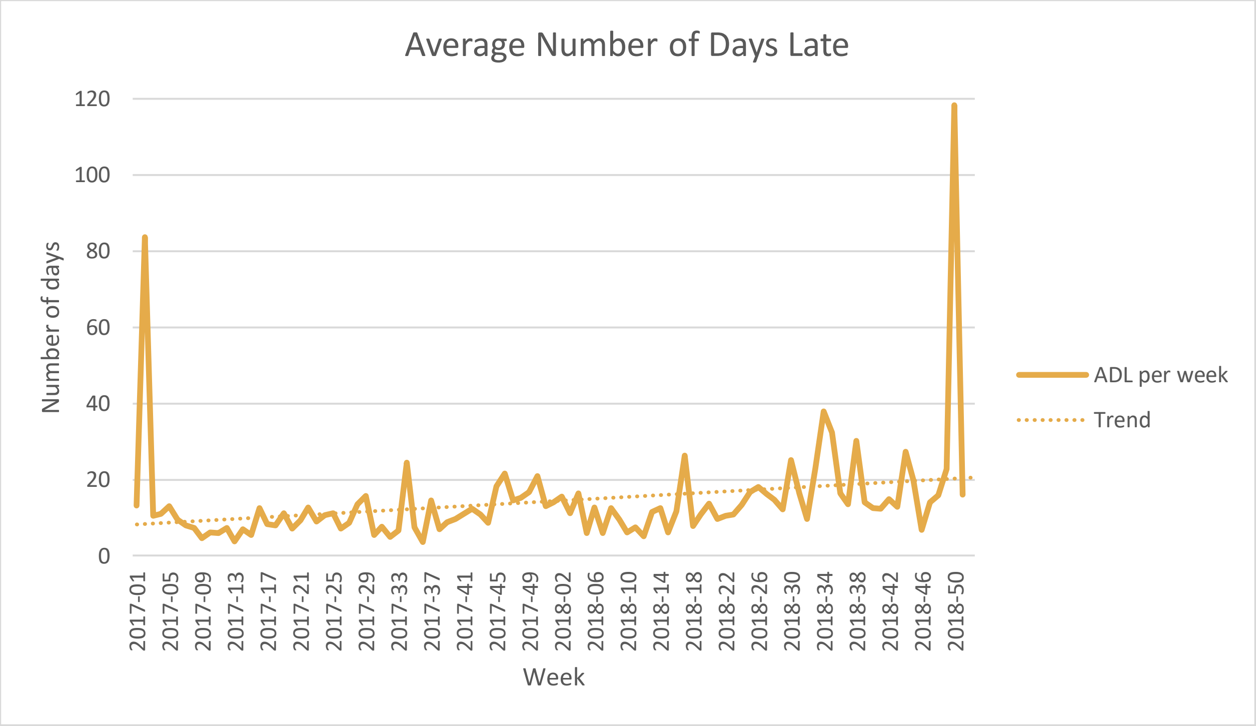


Figure 12: Average Number of Days Late

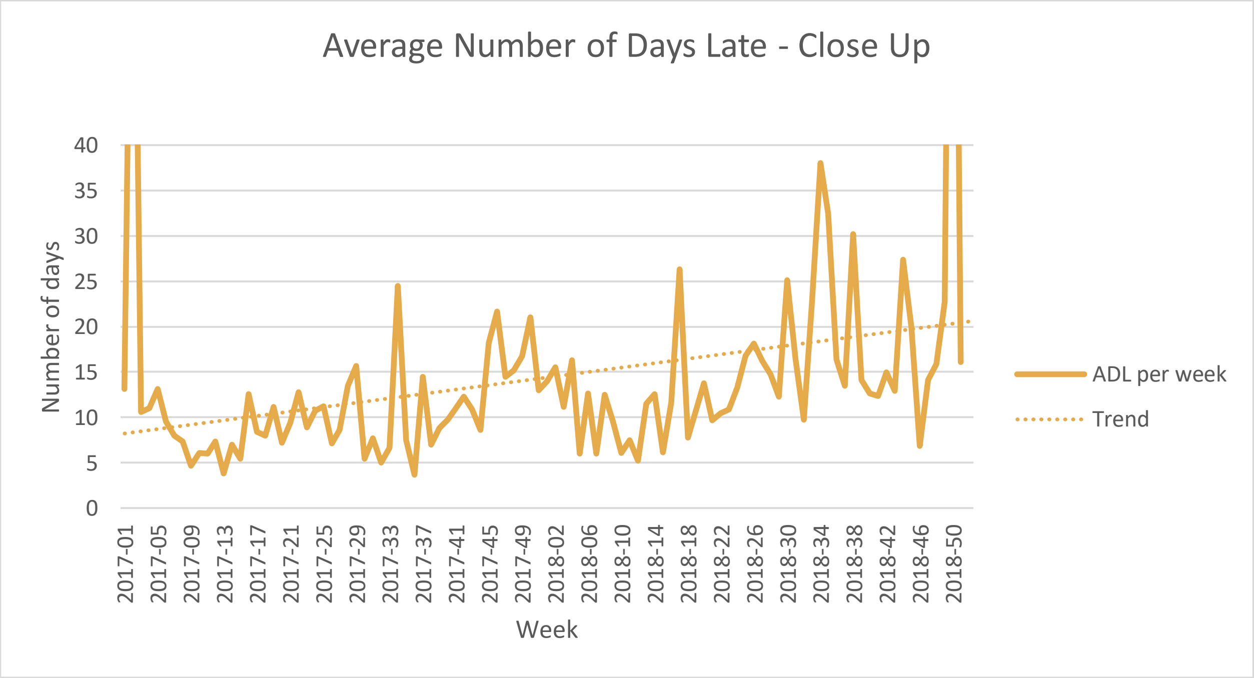


Figure 13: Average Number of Days Late - Close up

Average Set-Up Time

Whenever a worker starts processing a work order, this worker needs to keep track of how much time is spent on setting up. When setting up is finished, the worker registers the processing time. This is done via a digital screen on which the orders are displayed in the ERP system Glovia. The worker simply needs to press a button when he starts and press again when either setting up or processing is finished. We calculate the average set-up time (AST) based on this data. Not for all machines set-up times are relevant, for example straightening does not require any set-up activities. Also we are mainly interested in the sequence dependent set-up times. The set-up times that are dependent on the preceding and succeeding operation, provide an improvement opportunity. Because these set-up times are dependent on the sequence in which the orders are scheduled, these set-up times are interesting to include in the scheduling process.

$$Average\ set-up\ time = \frac{Sum\ of\ all\ set-up\ times}{Sum\ of\ all\ set-up\ and\ processing\ times} * 100\%$$

The AST is calculated as the percentage of hours spent on setting up of the total processing time required for Figure 14 shows the AST per month. The AST of 2018 is 8.4% of the total processing time.

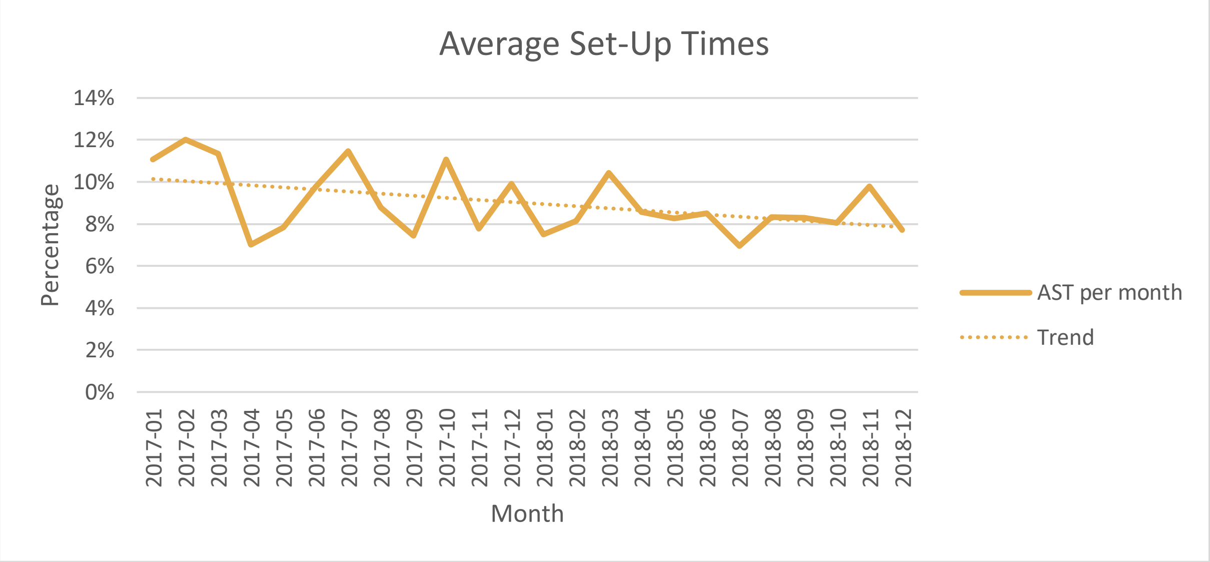


Figure 14: Average Set-Up Time

Average Queueing Time

The average queueing time (AQT) gives insight in how long an item has to wait before it will be processed by the next step. We define the AQT as the average queueing time in days per operation. An operation is one processing step of the routing of an item. The AQT is calculated as follows:

$$Average\ queueing\ time = \frac{Total\ queueing\ time}{Number\ of\ operations}$$

$$Queueing\ time = \ Start\ time - finish\ time\ of\ preceding\ operation\ of\ same\ job$$

Queueing time for operations that are preceded by processes that require waiting time, for example to cool down, are excluded. To calculate the AQT we use the registered processing times. Because this registration only starts when the first operation is started, we cannot calculate the queueing time for the first operation. This is also not relevant, because as long as the first operation has not started yet, there is no semi-finished product that has to wait.

Figure 15 depicts the AQT in days. The AQT of 2018 is 7.6 days. This is the baseline measurement of the AQT. We exclude the operations that are preceded by annealing and cleaning, because these processes require mandatory waiting time. The products need to cool down before they can proceed to the next production step.

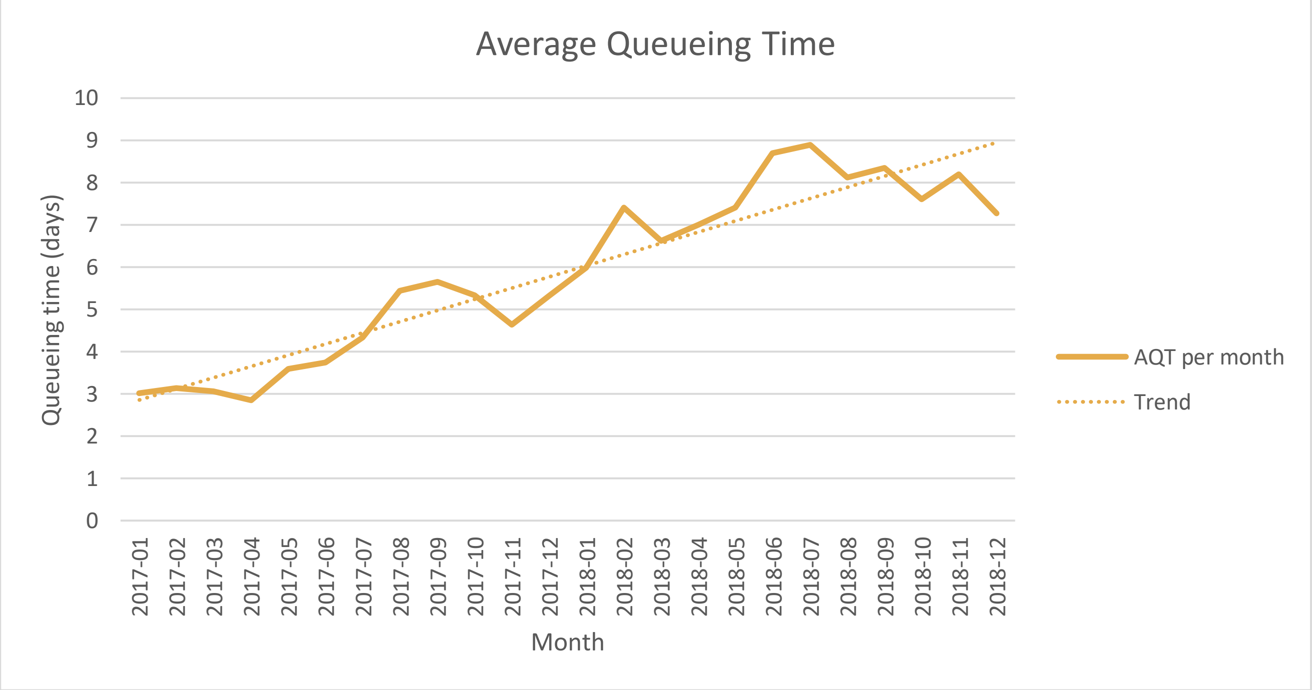


Figure 15: Average Queueing Time

2.3 Additional Scheduling Constraints

The on-time delivery performance highly depends on the actual available capacity. It especially depends on whether the scheduled capacity utilization complies with the actual available capacity, because the delivery date is based on the available capacity in Factory Planning. If this capacity deviates from the actual available capacity, the realization of work orders cannot be executed according to the schedule and orders will arrive too late at the customer. To get a realistic view of how much capacity is actually available, we consider several constraining factors that might be valuable for PM to include in the scheduling process. By “constraining” we mean that the effective capacity can never be higher than the most capacity-limiting factor allows. These constraints might however add up, if multiple constraining factors occur at non-coinciding times. We identify five factors that constrain the actual available capacity.

First, downtime of a machine reduces the actual available capacity of a machine. If the downtime is not incorporated in the capacity, more capacity than actually available might be utilized by the schedule. Second, the availability of workers with the required skills restricts the actual available capacity of a machine. This is not relevant for all machines, because some machines mainly run automatically. For many other machines a worker is required to keep the machine running. When no worker is available, the machine cannot be utilized. Third, the availability of tools and fixtures restricts the effective capacity of a machine if these are insufficient. Fourth, the availability of the required machine program can be restrictive if it is not well prepared. For most production operations, a machine needs a program in order to execute this operation. Before a machine can start processing, this program must be prepared. Fifth, when an order is released too late, production cannot start as planned, so at the scheduled time when this capacity was reserved, the machine is idle and when the order is finally released, this order needs to be processed later on, when the capacity is already reserved for another work order.

2.4 Conclusion

This chapter answers Sub Question 1 *"What is the performance of the currently generated schedules?"*. The processes at PM are structured as a job shop. The products that PM produces are partially very different and are produced in low volumes, this is because of the products that PM produces according to the ETO and MTO approach. PM however also produces MTS. This concerns the standard products. Products pass through the six departments each according to their own routing. The five production departments at PM are the cutting, drilling, milling, hardening and grinding department.

Orders are scheduled using Factory Planning. By setting several parameters, Factory Planning generates a schedule. Factory Planning is also used to determine the due dates for the customers. Factory Planning contains several scheduling extensions such as set-up time optimization, implementing priority rules and workforce scheduling. PM does currently not apply these extensions, because not all required parameters are known.

Relevant performance indicators in this research are the On Time Delivery Performance, Average Number of Days Late, Average Set-up Times and Average Queueing Time. The current performance of PM is based on data of 2018. PM currently has an OTDP of 80.3%. The trend is that this value is decreasing. The ADL of the orders shipped in 2018 is 17.6 days. This value is increasing over time. The AST of 2018 is 8.4% of the total processing time. The AST is decreasing. The current AQT at the production departments of PM is 7.6 days, and increasing over time.

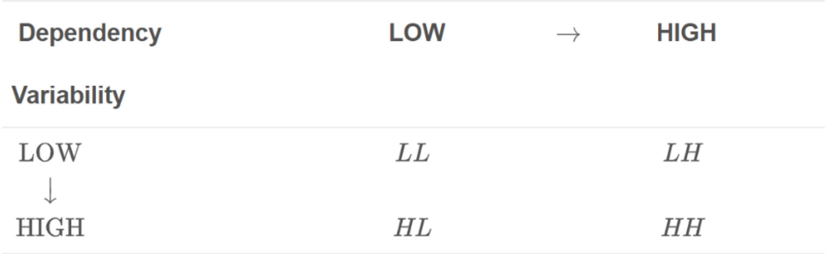
To create realizable schedules, it is valuable for PM to take five additional scheduling constraints into account. These additional scheduling constraints are machine downtime, worker availability, tool and fixture availability, machine program availability and on-time order release.

3 Literature Review

This chapter elaborates on the literature review that provides an answer to research question *"What methods for achieving an optimized production schedule in an MTO environment are described in literature?"*. Section 3.1 presents a framework for planning and control. Section 3.2 discusses the job shop scheduling problem and its extensions. Section 3.3 reviews the literature on one specific extension of the job shop scheduling problem, the dual resource constrained job shops. Section 3.4 elaborates on the various scheduling algorithms that literature presents. Section 3.5 concludes this chapter.

3.1 Make to Order Manufacturing Environments

In an MTO manufacturing environment it is a great challenge to make realistic production plans to achieve the due date promised to the customer. Providing reliable delivery dates is however an important customer service aspect. Besides reliable delivery dates, short lead times also provide a competitive advantage (Teo, Bhatnagar, & Graves, 2012). Hans et al. (2007) approach a job shop, which is a typical MTO manufacturing environment as a multi-project organization. A project is the whole package of activities that is required to meet the customer's request. For the products in this context that are customized, each project has unique elements. This customization also makes that the production process encounters variability. Especially for the products that are engineered to order, this variability is high, because no information on, for example, processing times is available beforehand. To make sure that projects will be finished on time, proper project planning is required. Project planning concerns the required resources and the objective of the project, as well as a project schedule. In most MTO companies however, multiple projects run in parallel, competing for the same resources. Hans et al. (2007) propose a positioning framework to categorize the various forms of multi-project companies. This categorization is done based on the variability and the dependability of projects. Figure 16 depicts this positioning framework. For MTO and ETO the LH and HH categories are relevant.



Dependency	LOW	→	HIGH
Variability			
LOW	LL		LH
↓			
HIGH	HL		HH

Figure 16: Positioning Framework for Multi-Project Companies (Hans, Herroelen, Leus, & Wullink, 2007)

Hans et al. (2007) also propose a hierarchical project planning-and-control framework, as can be seen in Figure 17. This framework distinguishes three hierarchical levels: strategic, tactical and operational. For each level, there are three functional planning areas: technological planning, capacity planning and material coordination. The type of objective and the time horizon differ among the managerial levels.

The technological planning area is concerned with the engineering process of the products. The resource capacity planning area manages that the available capacity and resources match the required capacity and resources. The material coordination concerns locating the right raw materials at the right location at the right time.

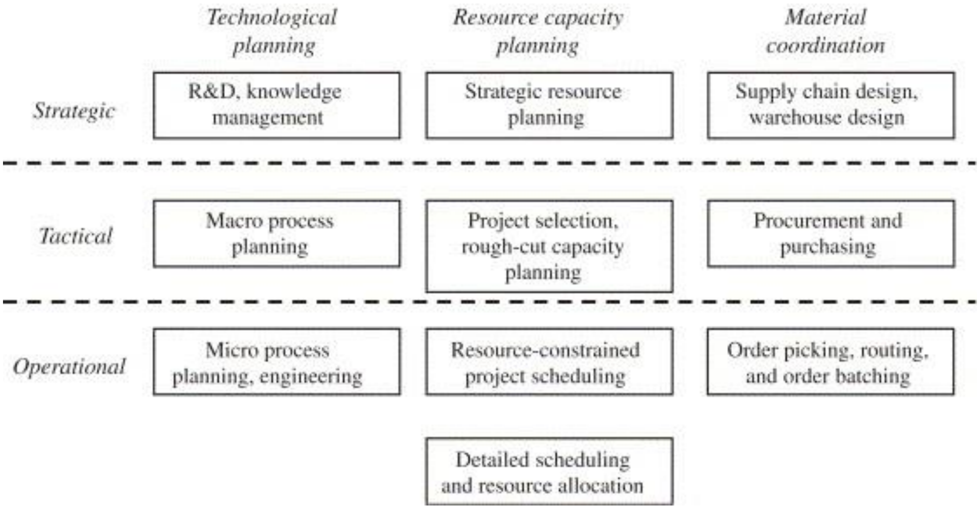

Technological planning
Resource capacity planning
Material coordination
Strategic
R&D, knowledge management
Strategic resource planning
Supply chain design, warehouse design
Tactical
Macro process planning
Project selection, rough-cut capacity planning
Procurement and purchasing
Operational
Micro process planning, engineering
Resource-constrained project scheduling
Order picking, routing, and order batching
Detailed scheduling and resource allocation

Figure 17: Hierarchical Framework for Planning and Control (Hans, Herroelen, Leus, & Wullink, 2007)

We are mostly interested in the resource capacity planning area. Resource capacity planning at the highest hierarchical level is about the question whether the current capacity and resources are sufficient for the demand that is expected on the long term. The definition of long term depends on the organization context. In the context of this research we can say long term is roughly about one year. Decisions such as machine procurement and hiring or firing employees are made at the strategic level. Decisions made at this level have a relatively high impact. Once strategic decisions are made, they cannot easily be adjusted on the short term. To make these decisions, aggregated information on customer demand and market behavior is required.

At the tactical level, decisions have less impact than at the strategic level and concern in this research context about one month. Decisions at the tactical level are about order acceptance and a rough capacity plan. Because this level is still concerned with planning quite some time into the future, the capacity is still flexible. For example decisions such as working overtime or subcontracting can be made to make sure the available capacity is sufficient. Companies tend to accept as many projects as possible and also try to promise a delivery date as early as possible. This will negatively influence the performance on the operational level if these decisions are not made after sufficiently assessing the impact on the capacity. This can lead to a poor delivery performance (Hans, Herroelen, Leus, & Wullink, 2007). To deal with this, Rough Cut Capacity Planning (RCCP) approaches are available. There are two RCCP variants, time driven RCCP and resource driven RCCP. Time driven RCCP considers the deadline as a constraint that must be met. In order to do so, non-regular capacity can be used. The objective of time driven RCCP is to minimize the cost of the total non-regular capacity used. Resource driven RCCP considers the regular available capacity as a constraint that must be met. No additional capacity can be realized. In the order scheduling process, the project lead time will be minimized (Gademan & Schutten, 2002).

At the lowest hierarchical level, the planning flexibility is very low, because the operational planning level is concerned with the short term planning. The length of the horizon that is defined as short term depends on the context. For PM it would be one week. In that case one week in advance orders will be scheduled in detail. This level is concerned with planning and control at the shop floor. The exact time, machine and worker, if relevant, are determined here. Typically, sequencing of the operations on specified machines is a task at the operational level. This resource capacity planning at the operational level is especially a challenge in MTO environments. The job shop layout in such environments makes it complex to create a schedule. The high number of different jobs, which each have a specific routing need to be assigned to various machines in a certain sequence. This results in a large number of possible schedules. Literature addresses this problem as the Job shop Scheduling Problem (JSP) (Zijm, 2000).

3.2 Job Shop Scheduling Problem

The current layout at PM is a job shop layout. For such layouts, there are plenty of schedule optimization methods described in the available literature.

In the classical JSP a set of jobs need to be processed by a set of machines. Each job consists of a certain number of operations, which need to be performed in a fixed order. Each operation also needs to be performed by a fixed machine. In modern manufacturing environments however, this dedication of operations to machines is not fixed. Instead, one operation type can be processed on more than one machine. This is called the Flexible Job shop Scheduling Problem (FJSP) (Ziaee, 2014). This problem is more complex than the classic JSP, which is proven to be NP-hard (Garey, Johnson, & Sethi, 1976). An additional problem that FJSP has to deal with, as opposed to the classic JSP is the assignment of operations to machines (Ziaee, 2014).

A problem that can be solved to optimality "quickly", i.e. in polynomial time, is known as a problem of the class P. Efficient solutions, i.e. exact algorithms can be thought of to solve such problems to optimality. Problems of the class P are part of a subclass of NP, which is Nondeterministic Polynomial. The problems that are in NP but not in P cannot be solved in polynomial time, but can be verified in polynomial time once a solution is found. These problems are called NP-problems. NP-problems can be solved to optimality by applying complete enumeration. Since the running time for solving NP-problems that are not in P increases exponentially when the problem size increases, only small problems can be solved to optimality (Fortnow, 2009). Another class of problems is the NP-hard class. If the subset P is not equal to the class NP, then the NP-hard problems cannot be solved to optimality in polynomial time. To provide a feasible solution to an NP-hard problem, generally heuristics are applied which try to find a solution as close to the optimal solution as possible (Lawler, Lenstra, Rinnooy Kan, & Shmoys, 1993). The JSP, or any variant of this problem, cannot be solved to optimality in polynomial time (Garey, Johnson, & Sethi, 1976). Various heuristics are proposed in literature that result in a feasible solution for the JSP and its variants.

The FJSP can be extended to the Dual Resource Constrained Flexible Job shop Scheduling Problem (DRCFJSP). Literature commonly considers, next to the constraining machine capacity, the workers as a constraining resource. The DRCFJSP consists of three sub-problems. One problem is that all operations need to be sequenced on the machine they are assigned to. We also see this sub-problem at the JSP. Besides that, the operations need to be assigned to the machines. This problem arises at the introduction of the FJSP. Also, operations need to be assigned to the workers. This problem comes about at the DRCFJSP (Dhiflaoui, Nouri, & Driss, 2018).

3.3 Dual Resource Constrained Job Shops

Most manufacturing systems in practice are not just constrained by machine capacity, but also by labor capacity. Workers in a manufacturing system represent the labor capacity. Not only the number of workers is of interest, but also the flexibility of the worker which is defined by the number of machines a worker can attend. This depends on the skill level of the worker. Both of the constraining resources, workers and machines need to be controlled (Thürer, 2018). In the remainder of this section we first present the problem definition of the DRCFJSP. Next we address literature on workers in scheduling problems. Finally, we review literature on scheduling methods in which often machine assignment is considered.

Problem definition

The purpose of solving the DRCFJSP is to schedule a number of jobs on a number of machines that need to be attended by a worker. Each job consists of a certain number of operations that need to be done in a determined order. These operations cannot be executed simultaneously, also an operation cannot start being processed before all its predecessors are finished. For each operation there is a subset of machines on which this operation is allowed to be processed. For each machine there is a subset of workers that can attend this machine. The progress of the job depends on the machine and worker availability. The time at which an operation can start depends on the finish time of its preceding operation, the finish time of the machine on which this operation is scheduled and the finish time of the worker that needs to attend this machine (Yazdani, Zandieh, Tavakkoli-Moghaddam, & Jolai, 2015).

Labor Capacity

If all workers have the same skill level, the workforce is homogeneous. This is not a realistic representation of most manufacturing companies in practice, because the capability to learn differs among workers. Also, workers that have more working experience will most likely have a higher skill level than workers that for example just started working. Besides, a manufacturing environment requires different skills for different tasks. So, the more skills a worker masters, the higher his or her flexibility is. A lot of researches experiment to find out the optimal flexibility level of workers (Xu, Xu, & Xie, 2011).

Kher & Fry (2001) show in their research that most due date performance improvement can be realized when the labor flexibility increases from one to two. A labor flexibility of two means that on average workers have the skill to operate two different machines. Kher & Fredendall (2004) confirm this observation. The actual impact of increasing the labor flexibility on performance measures such as flow time and tardiness depend on the interaction with other factors. An example of another factor is what dispatching rule is used.

Felan & Fry (2001) suggest a non-equal distribution of labor flexibility. By averaging the labor flexibility among all workers, a non-integer labor flexibility is possible. In their research they conclude that a labor flexibility of 1.7 performs just as well as the integer flexibility of 2. A lower flexibility is more desirable, because to achieve this, less training cost are required.

Another aspect of labor flexibility is the staffing level. The staffing level is the worker-machine ratio. ElMaraghy et al. (2000) found for a DRC manufacturing system in which the workers have a flexibility of two and where the busy time of the worker is equal to that of the machine, an optimal staffing level of 70%. If the staffing level would be increased, the marginal benefits are too little compared to the increase in staffing cost.

Since the DRC shop is dependent on the available labor capacity, it is interesting to optimize the assignment of workers to machines. The literature on this topic proposes two categories of assignment rules, which are known as the when-rules and where-rules. The first determines when a worker is available to move to another machine. The second determines to which machine the worker should transfer. Bobrowski & Park (1993) examine five when-rules and seven where-rules. In their research, Bobrowski & Park (1993) evaluate the performance based on the flow time and tardiness. They conclude that where-rules dominate any of the when- and dispatching rules presented in their research.

As opposed to the conclusions of Bobrowski & Park, Xu et al. (2011) conclude from their literature review that when-rules are more important to consider than where-rules. They claim that when-rules have a more significant effect on the overall performance of the system. In a more recent literature review, Thürer (2018) confirms this observation.

Little literature takes into account semi-automatic machines, i.e. machines that only require a worker during set-up and loading and unloading. Morinaga et al. (2014) consider setup-workers. Instead of assigning a worker to a machine the complete operation time of the operation, the worker is only necessary during set-up prior to the operating time. To generate a solution for production scheduling, this study uses a GA. Morinaga et al. (2014) execute several experiments to evaluate the performance on leveling the set-up load.

Machine Assignment

A lot of literature is available on controlling the machine capacity, i.e. scheduling a set of jobs on a set of machines, in a DRC environment. Since this problem is NP-hard, large problems cannot be solved to optimality in polynomial time (Garey, Johnson, & Sethi, 1976). The problem at hand at PM can be classified as a DRCFJSP. To solve scheduling problems, many methods are proposed in literature, although not always specifically applied to the DRCFJSP. To get an idea on the scheduling methods that are common in literature, we do not limit ourselves to literature that addresses the same problem definition as for PM, because this is hard to find.

Since the scheduling problem in the context of this research is NP-hard, often heuristics are proposed as a solution method. Silver (2004) defines heuristic as: *"a method which, on the basis of experience or judgement, seems likely to yield a reasonable solution to a problem, but which cannot be guaranteed to produce the mathematically optimal solution."*. Verbeeck (2011) discusses two categories of heuristics, namely single-pass and multi-pass heuristics. Single-pass heuristics pass through the steps of the heuristic, as the name suggests, once. This results in one solution. Multi-pass heuristics apply the heuristic steps multiple times. Every time the heuristic remembers the best found solution. This requires more computation time than single pass heuristics, but often results in better solutions. Heuristics are often, however, applicable to a specific problem.

3.4 Scheduling Algorithms and Heuristics

Silver (2004) discusses several basic heuristic types which are applicable to a broader range of problems. Amongst others, he discusses constructive methods. Constructive methods construct a solution step by step by using the information of the problem instance. The greedy method is a special constructive heuristic. The greedy method tries to get the most benefit as soon as possible. Another heuristic type that he discusses, contains the local search methods. This heuristic starts off with a feasible solution. By searching in the neighborhood of the current solution this type of heuristic tries to improve the current solution. The neighborhood can be defined in different ways, depending on the problem type. A neighbor can for example be reached by swapping two elements in the current solution. Local search heuristics can be applied to a solution found using constructive heuristics, to improve this solution. A drawback of local search techniques however, is that the process of searching for better solutions can get stuck in a local optimum. To escape from a local optimum, one can start over by generating a new initial solution or a more advanced procedure such as a metaheuristic is required. (Silver, 2004) A specific local search method is Steepest Descent, also known as Steepest Hill Climbing. Each iteration, this heuristic reviews the whole neighborhood and selects the neighbor that performs best of all neighbors. If this neighbor improves the current solution, this neighbor is selected. If no neighbor improves the current solution, a local optimum is reached and the search procedure stops. (Di Gaspero, 2003)

Over the past few decades, metaheuristics are upcoming as a method to solve, amongst others, scheduling problems. This is a third class besides the single-pass and multi-pass heuristics (Pellerin, Perrier, & Berthaut, 2019). A metaheuristic is a more advanced method that combines several heuristics and procedures and also includes local search (Verbeeck, 2011). Silver (2004) uses the definition of Osman (2003) for metaheuristics: *"A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to produce efficiently high-quality solutions. It may combine intelligently different concepts to explore the search space using adaptive learning strategies and structured information."*.

Literature proposes and discusses many metaheuristics for scheduling problems. Not all studies apply the metaheuristic to the DRCFJSP. In the following we discuss the metaheuristics that are well known and frequently discussed in literature. We discuss Tabu Search (TS), Simulated Annealing (SA), Variable Neighborhood Search (VNS), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Genetic Algorithm (GA).

TS is a combination of a constructive heuristic and an improvement heuristic. TS starts off with a feasible solution (constructive) and then looks for better feasible solutions (improvement). To escape from local optima, an inferior solution may be accepted. A tabu list keeps track of the previous found solutions, to prevent cyclic returning of the same solutions. The length of the tabu list is a controllable parameter (Silver, 2004). TS is well applicable to both discrete and continuous solution spaces and is also capable of dealing with large and more complex problems. Furthermore, TS consists of both a local and global search procedures. Due to this, TS is able to find a solution that outperforms the best solution found by local search procedures. Next to these strengths, TS encounters several weaknesses. A lot of parameters need to be determined and also the number of iterations can be very large. Furthermore, there is no theory formulated that supports the convergence behavior of TS. Also, to select the relevant parameters, one should have knowledge of the domain to which TS is applied. The data used must be structured efficiently to be able to apply the tabu list manipulation. Although this method is applicable to continuous search spaces, the neighborhood movements might however be difficult in continuous search spaces. Finally, an increasing number of objective functions makes it harder to design a good TS method (Zarandi, Asl, Sotudian, & Castillo, 2018).

SA also starts off with a feasible solution. Just like TS, SA also accepts inferior solutions. SA is however memoryless. The parameters that have to be selected for this method make up the cooling schedule. These parameters consist of among others, an initial temperature and the decreasing value of the temperature (Zarandi, Asl, Sotudian, & Castillo, 2018). As the temperature decreases the probability of accepting a solution that is inferior to the current solution also decreases (Silver, 2004). SA can be implemented in a parallel way when efficiency is emphasized. This method also has the ability to avoid getting stuck in local optima. Besides, the convergence of this method is theoretically proven. It is also very well capable of dealing with nonlinear models and many constraints. Other strengths are the flexibility and the versatility of the method. A weakness of SA is however that there is a clear tradeoff between solution quality and the required computation time, so when a higher solution quality is desired, the computation time severely increases (Zarandi, Asl, Sotudian, & Castillo, 2018).

VNS escapes from local optima by exploring multiple neighborhoods. Given a solution, VNS randomly chooses a solution from the first neighborhood. This solution is locally optimized and if this optimized solution is better than the previous than this solution is accepted. A next random solution is generated from the neighborhood of this new solution. If no new solution is found, the new random solution is generated from a broader neighborhood of this original solution. If the maximum neighborhood, determined on beforehand, is reached, the procedure stops (Silver, 2004).

ACO is based on the behavior of ant colonies that travel between their nest and food. The decision of what city to visit next depends on two factors: the distance and the probability that this city is visible. The visibility depends on the amount of pheromones of the ants that have travelled along that city. Once each ant of the colony has chosen a tour, the amount of pheromones at each city is updated. The two controllable parameters of this metaheuristic are the number of ants and how many pheromones each ant lays down (Silver, 2004). The ACO method uses positive feedback to rapidly discover good solutions. Also, diversity comes about in an emergent way, so no explicit mechanism is required to ensure diversity. This method is suited to be applied to dynamic problems. ACO has the ability to efficiently solve discrete problems. For this heuristic convergence is theoretically proven. As opposed to these strengths, ACO encounters several weaknesses. The random decisions are not independent. This method is not effective in solving continuous problems (Zarandi, Asl, Sotudian, & Castillo, 2018).

PSO is easy to use. It is very well applicable to continuous problems and it has a high convergence rate. As opposed to the other metaheuristics we address, the parameters to be determined are little and they are easily implemented. PSO can deal with dynamic environments and is fast in solving nonlinear problems. Computation cost are low. The local search ability is however weak, it easily gets stuck in a local optimum (Zarandi, Asl, Sotudian, & Castillo, 2018).

Evolutionary algorithms are a type of metaheuristic that try to reflect natural evolutionary processes. This type of metaheuristics is also known as genetic algorithms (GAs). These algorithms use a group of solutions. Each solution that is present in the group is evaluated to select a subset at each iteration. The first group of solutions can be randomly generated, or a constructive heuristic can be used. For the selection phase individual solutions can be randomly selected or in a deterministic manner. Both use the fitness value of the solution in the selection process. To produce individuals for the next generation, two solutions are combined by crossover. GAs consist of various parameters, a few of which are population size, the mechanism that generates the size of the mutation and the probability an individual gene mutates. GAs have the strength of being able to solve continuous problems in an efficient way. Also, they are easy to implement and are less likely to get stuck in a local optimum, because multiple individuals represent different solutions in the search space at the same time. Furthermore, in the process of exploring the search space, the partial solution that have already been found are not completely lost. A wide range of solutions is guaranteed due to the random mutation. The weaknesses of GAs are however that such a method may fail to find any satisfactory solution. Also, computation cost are high and premature convergence might occur. In the implementation of a GA it is a challenge to determine the best parameters. Finally, GAs require the problem to be encoded in a chromosome. Although this depends on the problem, this can be a difficult process (Zarandi, Asl, Sotudian, & Castillo, 2018).

Over the past few decades, several reviews are executed on the different methods for scheduling. Hartmann & Kolisch (2000) execute an experimental evaluation of, among others, metaheuristics that they found in the, back then, current literature. Later, Kolisch & Hartmann (2006) update their evaluation in which they include the newly proposed solution methods. More recently Pellerin et al.

(2019) execute a survey focused on hybrid metaheuristics. They experiment according to the same experimental protocol as followed by Hartmann & Kolisch and Kolisch and Hartmann, in 2000 and 2006 respectively. This makes it possible to compare the results. Pellerin et al. find 36 hybrid metaheuristics that outperform the best performing approaches described in the review of Kolisch & Hartmann in 2006. A common method among these 36 best performing metaheuristics is the GA approach. This observation is in line with the research of Çaliş & Bulkan (2015).

Çaliş & Bulkan (2015) review the scheduling methods presented in literature on job shop scheduling problems. They observe that GA is the most used scheduling method. In 26.4% of the studies that Çaliş & Bulkan include in their review, GA is the applied scheduling method. The second most frequent method is the method that makes use of neural networks. This method is applied in 18.9% of the studies. The other methods that Çaliş & Bulkan include are Beam Search, TS, Agent Based Systems, ACO, PSO, VNS, Fuzzy Logic and Bee Colony Optimization.

Amjad et al. (2018) make the same conclusion for FJSP. They state that *"GA has proven to be one of the most effective evolutionary techniques for solving (...) FJSP"*. In the timeframe 2001 to 2017 Amjad et al. observe that hybrid GAs are more popular than the pure GA.

Akbar & Irohara (2018) propose a permutation-based GA (PGA) to solve the DRC scheduling problem. They encounter semi-automatic machines which can run partially without a worker being present. The objective is to minimize the makespan. To test the performance of their proposed PGA, Akbar & Irohara compare this method to a mixed integer lineair programming (MILP) model, to which a solver is applied and random search. The results show that the PGA can solve the DRC scheduling problem in a reasonable time. PGA can solve faster than the solver which finds a solution of good quality for the MILP model and at the same time provides a better solution than random search. This research considers identical parallel machines. The authors propose for further research to extend the model to, amongst others, a job shop setting.

Yazdani et al. (2015) address the DRCFJSP. In the aim of minimizing the makespan, they develop two meta-heuristic, SA and Vibration Damping Optimization (VDO). This research concludes that VDO performs better than their SA. Yasdani et al. (2015) propose for further research to develop hybrid meta heuristics to solve the DRCFJSP. Such a hybrid meta heuristic can for example be a combination of VNS and SA or GA and VNS. Also they propose to consider sequence dependent set-up times, which is an important factor in the DRCFJSP.

Like Yazdani et al. (2015), Wu et al. (2018) also address the DRCFJSP. Wu et al. (2018) propose a hybrid GA. To improve the convergence speed of the GA, they integrate VNS as local search method in the GA. This method shows promising results. The proposed GA-VNS algorithm outperforms GA without VNS as local search and the Hybrid Discrete Particle Swarm Optimization. The objective in this research is to minimize the makespan.

The objective of minimizing the makespan is a popular objective in literature. Minimizing the makespan creates however dense schedules. This makes these schedules sensitive to any disturbance that might happen on the shop floor (Al-Hinai & ElMekkawy, 2011). There are numerous events that cause disturbances on the shop floor. Paprocka et al. (2017) distinguish three categories of disturbances. Disturbances can be related to resource availability, such as machine failure, unavailable materials or worker absenteeism. Also disturbances can be related to orders, for example a rush order can come in or rework needs to be done. Finally disturbances can be related to errors in the production parameters such as the estimated processing times. To deal with these types of disturbances, robustness and stability are important aspects of a production schedule. Liu et al. (2007) define a robust schedule as follows: *"A schedule is robust if its performance degrades a small degree under disruptions."* According to Liu et al. *"a schedule is stable if the deviation that includes time deviation or sequence deviation is very small between the predictive and the realized schedule"*. A stable schedule is also called a flexible schedule (Al-Hinai & ElMekkawy, 2011).

To achieve a robust production schedule slack can be encountered in the schedule. Slack is buffer time that provides room for operations to be delayed without influencing the final time the job is finished. By using lateness as the objective to optimize, earliness is also detected. This earliness creates a buffer, also called slack time which improves the robustness against unforeseen disturbances of a schedule (Jensen, 2001).

3.5 Conclusion

This chapter answers Sub Question 2: *"What methods for achieving an optimized production schedule in an MTO environment are described in literature?"*.

If we position the problem at hand in the hierarchical framework for planning and control in Figure 17 we see that the area of concern is the operational resource capacity planning. This includes resource-constrained project scheduling and detailed scheduling and resource allocation. The operational planning level is concerned with the short term planning. About a week in advance orders are scheduled in detail. This level is concerned with planning and control at the shop floor. The exact time, machine and worker, if relevant, are determined here. Typically, sequencing of the operations on specified machines is a task at the operational level.

The job shop layout in MTO environments makes it complex to create a schedule. The high number of different jobs, which each have a specific routing need to be assigned to various machines in a certain sequence. This results in a large number of possible schedules. Literature addresses this problem as the FJSP. To find a scheduling solution that fits the needs of PM, we focus on the DRCFJSP. For these kind of problems both worker assignment and machine assignment are considered in the scheduling process. We deviate from the way literature addresses the worker assignment, because an additional point of interest at PM is simultaneously assigning multiple machines to one worker, which is possible for the machines that run (semi)automatically.

There are plenty of problem-solving methods in general. Especially metaheuristics are of interest, because they are very well capable of dealing with large problems which are NP-hard. Common metaheuristics that are applicable to the JSP problem are TS, SA, ACO, GA and PSO. In some cases, hybrid metaheuristics can outperform traditional metaheuristics. It is hard to say what the best scheduling approach is for PM. We observe that GA is a very common approach in literature for various scheduling problems. We expect that GA is capable of generating improved schedules for PM. To improve the local search ability of the GA, we are interested in the hybrid metaheuristic GA-VNS. An interesting addition is the robustness of the schedule. This can be achieved by minimizing the lateness.

4 Improving Factory Planning

To provide insight for PM on how to improve the production schedules, this chapter and Chapter 5 answer Sub Question 3: *“What are potential scheduling improvements for PM?”*. In answering this question, we investigate two approaches to solve the scheduling problem. The first approach considers the planning system that PM currently uses. This system, Factory Planning, consists of several extensions that can be implemented additionally to the current way of using Factory Planning. Sub Question 3a addresses this approach and is formulated as**:** *“How can existing extensions of the planning system of PM be implemented?”*. This chapter answers Sub Question 3a. The second approach is to create a new scheduling process. To improve the scheduling process, we set up a scheduling algorithm. Sub Question 3b elaborates on this second approach: *“What should a new and optimal scheduling process for PM look like?”*. Chapter 5 answers this sub question. After discussing the Factory Planning extensions, we also address Sub Question 4: *“What is the impact of the scheduling improvements for PM?”*.

Section 4.1 discusses the extensions we expect to improve the schedules generated by Factory Planning. Section 4.2 describes how we implement these extensions in Factory Planning. Section 4.3 represents the results of applying the Factory Planning extensions to the production schedule. Section 4.4 concludes this chapter.

4.1 Factory Planning Extensions

Recall from Chapter 1 that we identify two main causes that are related to the scheduling process that result in a too low OTDP. These are that set-up optimization is not considered in the scheduling process and that worker capacity is not taken into account as a second resource constraint, besides the machine capacity. To improve the performance of the generated schedules, ideally we would apply these two extensions to Factory Planning. Due to time constraints we focus on extending Factory planning by the set-up optimization. For this extension we are able to implement a small set of data to actually run the optimization for a small section of the schedule. For the worker resource constraint this is more complex, because when we decide to include workers, which we expect to have quite an impact on the lead time since an constraint is added, it is important to get all base data right. Since this data is not yet collected for the this Factory Planning Extension, we decide to leave this for further research, because for now collecting all relevant data in detail is too time consuming. We do however consider this additional resource constraint in Chapter 5, where we investigate the impact of including this constraint. For this purpose we collect worker data at PM on a high detail level. In the following we elaborate on the set-up optimization extension.

Set-up Optimization

The idea of set-up optimization is that scheduling operations of items that have the same characteristics one after another, without being interrupted by an operation that processes an item that has different characteristics, will reduce the time needed to set up the machine. This reduces the required capacity of this machine, because the set-up time for the second scheduled operation is reduced. So, we expect set-up optimization to reduce the required capacity of the machines, which in turn creates more room, in terms of machine capacity, to finish jobs earlier and thus reduces the lead time for the customer.

The required set-up time depends on the characteristics of both the predecessor and the successor item of two consecutive operations on one machine. Literature addresses this phenomenon as sequence dependent set-up times (Naderi, Zandieh, & Fatemi Ghomi, 2009). If the characteristics of both items are the same for all the characteristics that are relevant to set-up time, then no set-up time is needed when a machine switches from one operation to its successor. When one or multiple characteristics are not the same for two consecutive operations, then set-up time is required. How long this set-up time is, depends on what and how many characteristics differ between the items of two consecutive operations. We call the characteristics that influence the set-up time set-up parameters. Each item has a certain value for each set-up parameter. In general, set-up time can for example be impacted by the set-up parameter 'Color'. In that case an item can, for example have the parameter value 'black' or 'green'.

When Factory Planning executes the set-up optimization, Factory Planning tries to combine operations in such a way that the required set-up time is minimized. The user can define up to five different set-up parameters for each work center. The selected set-up parameter types are applied in hierarchical order, which means that the operations are first combined based on the first parameter type, then based on the second parameter type and so on. For each machine the user can select up to five parameter types from the list of parameter types that is linked to the relevant work center. For each item the user can define a value for each set-up parameter that is defined for any work center. This method makes sense for PM for the most common and standard items, because the more set-up parameters that correspond between two consecutive operations on one machine, the lesser set-up time is required. In reality it might however be hard to define a true hierarchy between the various set-up parameters. It might occur that the parameter value on the first set-up parameter does not correspond, but the second does. In this case it is likely that the set-up time does reduce, but Factory Planning does not detect this because the first set-up parameter did not match. So, this set-up optimization of Factory Planning works as long as there is a true hierarchy in the set-up parameters. Because of this we think the application of Factory Planning is rather limited for PM.

Factory Planning contains three predefined set-up parameters, which are 'First Material', 'First Tool' and 'Production Item'. These set-up parameters have the following logic. If 'First Material' is selected for the set-up optimization, Factory Planning checks the BOMs of all operations that are waiting and groups the operations that have the same first item. The 'First Tool' parameter combines all items with the same first tool in the BOMs. If 'Production Item' is selected, Factory Planning clusters orders that are part of the same final product. We think there are set-up parameters that are more suitable for PM. So, we do not consider these three set-up parameters. These three parameters are too limited, because in practice at PM these three parameters are not the most important factors that determine the required set-up time.

It is very complex and time consuming to gather all the data that is required to fully apply this extension to Factory Planning. Therefore we execute experiments in which we implement the set-up optimization on a small scale to be able to evaluate the effect of this extension. Based on the experience of the foremen at the production departments at PM, we make the following decisions on what products and production process to encounter in our experiments.

PM produces, amongst others, various types of linear bearings. Figure 18 shows an RSDE Linear Bearing Set. This is an item for which a lot of set-up time can be saved if set-up optimization is applied. Also, this is an item that is not too complex and is part of the standard assortment of PM. Since it is a rather complex task to determine how much set-up time is required for each combination of set-up parameters we focus, for the purpose of this research, on the roll diameter of the standard linear bearings. So, the set-up parameter is the 'Roll Diameter'. The Roll Diameter value can be 1.5, 2, 3, 4, 6, 9, 12 or 15. These values denote the roll diameter in millimetres. To keep the amount of work to implement the set-up optimization small, but at the same time be able to see significant effect of the set-up optimization, we only include 1.5, 2 and 3 as roll diameter for the linear bearings. These are the most common sizes, so in this way we catch most items without making the experiment too large.

Figure 18: RSDE Linear Bearing Set (PM, 2019c)

Set-up optimization is not relevant for all machines. Only machines that are concerned with changing the machine settings depending on the item that is going to be processed are relevant for the set-up optimization. For example the machines of the hardening department do not encounter set-up time depending on the parameter values of the items that need to be hardened. The work center that is mostly concerned with changing over the machine settings in relation to the linear bearings, i.e. set-up, is the drilling department. The drilling department has multiple Fanuc machines that are required to process one of the operations of the linear bearings. There are however slight differences between these machines which again makes set-up optimization more complex. For the purpose of our experiment we first want to focus on one machine. According to the foreman of the drilling department Machine 4 is most suitable. Further specifications of this machine are not relevant for this research.

To summarize, for this Factory Planning extension, we execute experiments to review the effect of this extension. For these experiments we include a set of standard linear bearings. The set-up parameter is the roll diameter of the linear bearings. We apply set-up optimization only to Machine 4 which is a machine at the drilling department.

4.2 Application of Set-up Optimization

To determine the impact of the set-up optimization in Factory Planning we set up an experiment. The relevant KPI on which the impact will be assessed is the total set-up time. We compare the required set-up time when set-up optimization is enabled with the required set-up time when set-up optimization is not enabled. The latter is the original situation. We expect the total set-up time required to reduce when set-up optimization is enabled. Due to the set-up optimization, realized due dates might become delayed, because Factory Planning tries to match orders with the same characteristics, such that the required set-up time will become shorter. This might mean that orders that have a due date further in the future will be placed closer to the present, and consequently push orders with an earlier due date further into the future. To prevent this effect from becoming too large, certain parameters can be set. These parameters are "Optimization Range", "Time Limit" and "MRP start date for set-up optimization".

The optimization range is the time period in which set-up optimization takes place. The number of days entered here added to the day on which the scheduling process is initiated, the planning horizon start, determines the end of the set-up optimization horizon. During this period, operations are scheduled for which the set-up is optimized, see Figure 19.

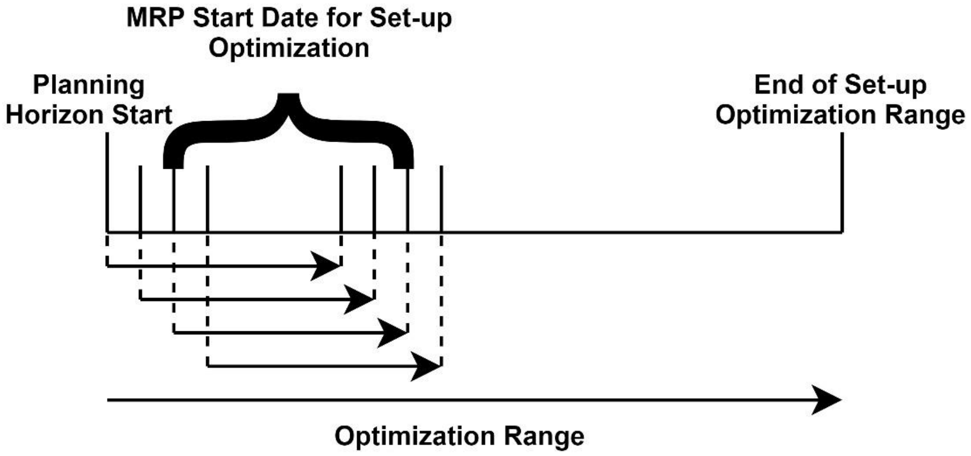

MRP Start Date for Set-up Optimization
Planning Horizon Start
End of Set-up Optimization Range
Optimization Range

Figure 19: Set-up Optimization Parameters

The optimization time limit defines how many consecutive days operations with the same characteristics are scheduled. There is no reason why this should be limited for PM, so we set this value equal to the Optimization range value. A reason to consider a restricting value for the optimization time limit could for example be relevant in an environment that uses paint. By limiting the number of consecutive days operations that require the same colour of paint are scheduled, prevents the other paint colours from drying out.

The horizon MRP start date for set-up optimization defines how far in the future Factory Planning should look to match an operation to the operation that is currently being scheduled. Figure 19 depicts this as smaller, overlapping time buckets within the optimization range. To keep the figure readable, Figure 19 shows only four of these time buckets, but in reality this process continues until the end of the set-up optimization range. This value can especially manage the trade-off between set-up minimization and the on-time delivery performance. This is an interesting setting to experiment with to find the optimal value. To answer the question what the best settings are for the set-up optimization we run several experiments. Table 1 shows the experiments we run in Factory Planning. For the optimization range we try a long, medium and short time span. It does not make sense to set the MRP start date as a larger value than the optimization range, because the optimization range is already limiting for what time span set-up optimization is applied.

Experiment Nr.	Set-up Optimization	Optimization Range	MRP Start Date
1		N/A	N/A
2	✓	1000	1000
3	✓	1000	30
4	✓	1000	10
5	✓	1000	2
6	✓	30	30
7	✓	30	10
8	✓	30	2
9	✓	2	2

Table 1: Factory Planning Experiments

For the set-up optimization we use the input data as Table 2 depicts. For this small scaled implementation of set-up optimization we apply set-up optimization to Machine 4. When an item has the same parameter value as the preceding item on the same machine the setup time is 0. When the parameter values do not match, the set-up time is 0.25 hours.

Machine Nr.	Parameter Value from	Parameter Value to	Set-up Time (hours)
Machine 4	1.5	1.5	0
Machine 4	2	2	0
Machine 4	3	3	0
Machine 4	1.5	2	0.25
Machine 4	1.5	3	0.25
Machine 4	2	1.5	0.25
Machine 4	2	3	0.25
Machine 4	3	1.5	0.25
Machine 4	3	2	0.25

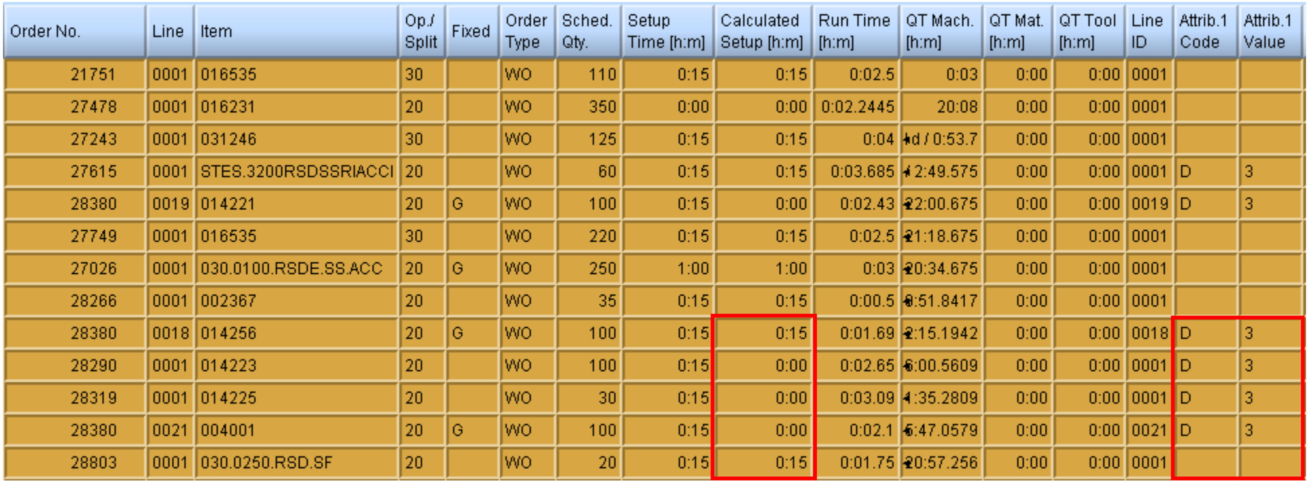
Table 2: Set-up Optimization Data

We analyze the results using the output data of Factory Planning. We let Factory Planning schedule the current set of orders that is available in Factory Planning up to the end of 2019.

We have to note that the experimenting possibilities of Factory Planning are limited. Factory Planning is partially a black box, so we cannot model this scheduling algorithm to execute various experiments in another environment. Factory Planning provides the opportunity to create schedules offline. This is the test environment of Factory Planning. Unfortunately this environment can be used by multiple users simultaneously. The same test environment is also used by the colleagues at the location in Hengelo, so it is not convenient to temporarily exclude other user from the Factory Planning test environment. Therefore we cannot guarantee that nobody interferes with our experiment. For example for the case where we want to compare the effect of set-up optimization we observe that for the second scheduling run where we applied the set-up optimization orders were added to the set in the test environment of Factory Planning. This makes it hard to compare the performance of various scenarios.

4.3 Results of Factory Planning Extensions

Due to the limited experimenting possibilities in Factory Planning we did not succeed to get the results for the experiments described in Table 1. We can however see that set-up optimization decreases the total required set-up time. Figure 20 shows a small section of the operation list of Machine 4 after set-up optimization. Appendix B: Set-up Optimization Results shows the complete operation lists of Machine 4 for both the case where we do, Table B2, and do not include set-up optimization, Table B1. For the case where we do include set-up optimization, we take an optimization range and an MRP start date of 1000 days. An operation list is the list that shows all operations that are scheduled on one machine. The red frames in Figure 20 show that when the set-up parameter values are the same for two succeeding operations the set-up time gets neglected. In this case there are four succeeding operations with a roll diameter (D) of 3 millimetres. Only the first of these operations gets accounted set-up time. The following three do not require set-up time so this set-up time is adjusted, based on the input in Table 2, to 0.



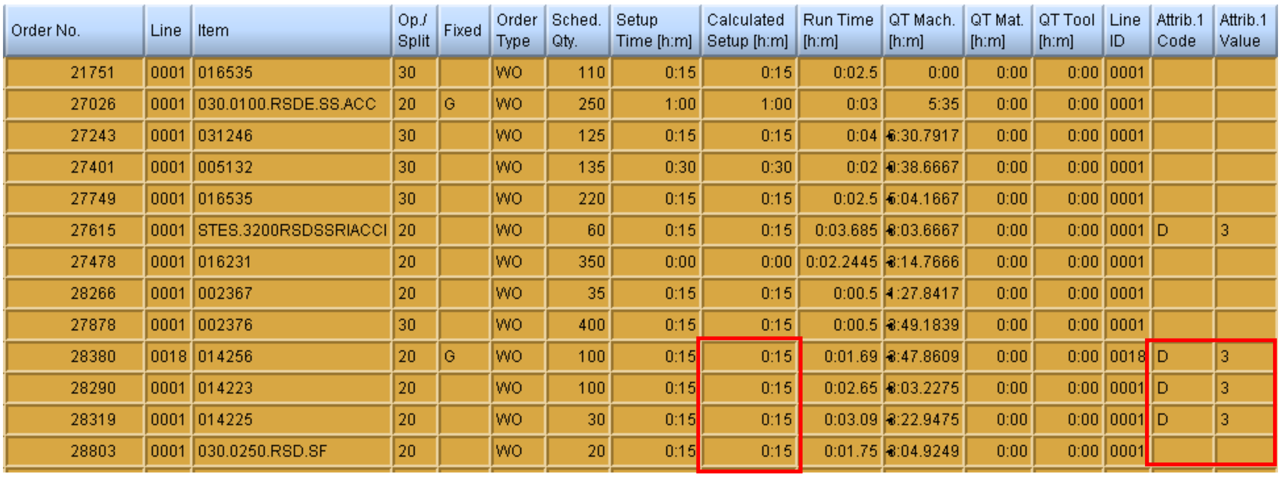
Order No.	Line	Item	Op./ Split	Fixed	Order Type	Sched. Qty.	Setup Time [h:m]	Calculated Setup [h:m]	Run Time [h:m]	QT Mach. [h:m]	QT Mat. [h:m]	QT Tool [h:m]	Line ID	Attrib.1 Code	Attrib.1 Value
21751	0001	016535	30		WO	110	0:15	0:15	0:02.5	0:03	0:00	0:00	0001		
27478	0001	016231	20		WO	350	0:00	0:00	0:02.2445	20:08	0:00	0:00	0001		
27243	0001	031246	30		WO	125	0:15	0:15	0:04	d / 0:53.7	0:00	0:00	0001		
27615	0001	STES.3200RSDSSRIACCI	20		WO	60	0:15	0:15	0:03.685	12:49.575	0:00	0:00	0001	D	3
28380	0019	014221	20	G	WO	100	0:15	0:00	0:02.43	22:00.675	0:00	0:00	0019	D	3
27749	0001	016535	30		WO	220	0:15	0:15	0:02.5	21:18.675	0:00	0:00	0001		
27026	0001	030.0100.RSDE.SS.ACC	20	G	WO	250	1:00	1:00	0:03	20:34.675	0:00	0:00	0001		
28266	0001	002367	20		WO	35	0:15	0:15	0:00.5	8:51.8417	0:00	0:00	0001		
28380	0018	014256	20	G	WO	100	0:15	0:15	0:01.69	2:15.1942	0:00	0:00	0018	D	3
28290	0001	014223	20		WO	100	0:15	0:00	0:02.65	6:00.5609	0:00	0:00	0001	D	3
28319	0001	014225	20		WO	30	0:15	0:00	0:03.09	4:35.2809	0:00	0:00	0001	D	3
28380	0021	004001	20	G	WO	100	0:15	0:00	0:02.1	6:47.0579	0:00	0:00	0021	D	3
28803	0001	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	20:57.256	0:00	0:00	0001		

Figure 20: Operation List Set-up Optimization

From the experiments we intend to execute we find a 16.5% reduction in required set-up time for Experiment 2, which includes set-up optimization as opposed to Experiment 1 which does not include set-up optimization. The values of the Total Delay and Number of Orders Late which Factory Planning calculates do however not change. This is not in line with our expectations. We also observe while running the experiments in Factory Planning, that applying different values for the Optimization Range and the MRP Start Date do not impact the results. The Total Delay, as calculated by Factory Planning, does not change when we increase the MRP Start Date. We expected that applying the set-up optimization to a larger time interval, indicated by the MRP Start Date value would increase the total delay and the number of orders late, because by matching operations and thus pulling them closer to the current date of the schedule will push other operations to the back. This negative impact that we expected is however limited. We expect this limitation to be the result of the precedence constraint. The impact of set-up optimization is limited because the operations scheduled on this machine cannot be scheduled before their preceding operation which is scheduled on another machine. Also, the lead times become shorter because the total set-up time is reduced, which also restricts the negative impact of the set-up optimization on the delay.

When comparing the set-up time required when we apply set-up optimization to when we do not apply set-up time the results are biased. In the case when Factory Planning does not apply set up time optimization it sometimes occurs that two operations that succeed each other do by coincidence have the same set-up parameter values. In this case Factory Planning does not detect this and thus includes the set-up time for both operations while in practice this set-up time will not be required. Thus the reduction in required set-up time when set-up optimization is applied will in reality be lower. To illustrate, Figure 21, displays part of the operation list of Machine 4 that does not include set-up optimization. The red frames indicate the operations with the same set-up parameter value that coincidentally follow each other. We see that set-up time is still included in the schedule.

If we account for this set-up optimization that occurs by chance the impact of including set-up optimization is smaller. Instead of an 16.5% reduction of set-up time, the reduction is only 9.5%. For this specific example we see that 7.7% of the set-up time is overestimated due to set-up time reduction by coincidence. This is an interesting observations even when set-up optimization is not considered. This means that the calculated set-up times do not represent reality accurately. When creating the schedule Factory Planning accounts for the set-up time that is defined for each operation. Because coincidental set-up time optimization is not considered, more time is scheduled for these operations, while in practice this set-up time is not required on the shop floor. This results in being finished earlier



Order No.	Line	Item	Op./ Split	Fixed	Order Type	Sched. Qty.	Setup Time [h:m]	Calculated Setup [h:m]	Run Time [h:m]	QT Mach. [h:m]	QT Mat. [h:m]	QT Tool [h:m]	Line ID	Attrib.1 Code	Attrib.1 Value
21751	0001	016535	30		WO	110	0:15	0:15	0:02.5	0:00	0:00	0:00	0001		
27026	0001	030.0100.RSDE.SS.ACC	20	G	WO	250	1:00	1:00	0:03	5:35	0:00	0:00	0001		
27243	0001	031246	30		WO	125	0:15	0:15	0:04	6:30.7917	0:00	0:00	0001		
27401	0001	005132	30		WO	135	0:30	0:30	0:02	0:38.6667	0:00	0:00	0001		
27749	0001	016535	30		WO	220	0:15	0:15	0:02.5	6:04.1667	0:00	0:00	0001		
27615	0001	STES.3200RSDSSRIACCI	20		WO	60	0:15	0:15	0:03.685	8:03.6667	0:00	0:00	0001	D	3
27478	0001	016231	20		WO	350	0:00	0:00	0:02.2445	3:14.7666	0:00	0:00	0001		
28266	0001	002367	20		WO	35	0:15	0:15	0:00.5	4:27.8417	0:00	0:00	0001		
27878	0001	002376	30		WO	400	0:15	0:15	0:00.5	3:49.1839	0:00	0:00	0001		
28380	0018	014256	20	G	WO	100	0:15	0:15	0:01.69	3:47.8609	0:00	0:00	0018	D	3
28290	0001	014223	20		WO	100	0:15	0:15	0:02.65	8:03.2275	0:00	0:00	0001	D	3
28319	0001	014225	20		WO	30	0:15	0:15	0:03.09	3:22.9475	0:00	0:00	0001	D	3
28803	0001	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	3:04.9249	0:00	0:00	0001		

Figure 21: Operation List Without Set-up Optimization

than according to the schedule, the semi-finished product will be sent to the next department for its next operation, while this machine might not be available yet for this operation. If this effect is too large, these events might cause disturbances and confusion on the shop floor which in turn could lead to reduced efficiency of the workers.

4.4 Conclusion

This chapter answers Sub Question 3a: "*How can existing extensions of the planning system of PM be implemented?*" and Sub Question 4 in relation to the Factory Planning extensions: *"What is the impact of the scheduling improvements for PM?"*.

Set-up optimization is a limited Factory Planning extension. The procedure of the set-up optimization in Factory Planning might not be suitable for all items, because set-up optimization follows a hierarchically method to determine the set-up time when two items are matched based on one or multiple set-up parameters. Set-up optimization is however capable of reducing the lead times for the items that are suitable for this extension. Not all items might be suitable for set-up optimization, especially the highly customized items that are ordered infrequently. Set-up optimization can however be implemented gradually. This is an important feature because PM can start implementing set-up optimization by collection set-up time data on standard items. This will already result in reduced product lead times and a more accurate representation of reality. The latter is important to be able to realize the schedules and thus be able to delivery on time.

For a small set of items and only applying set-up optimization to one machine in the set-up optimization we already see a set-up time reduction when we apply set-up optimization. Even though the set-up time reduction is in fact less than if we just compare the results from set-up optimization to the results of no set-up optimization, there still is an improvement. We expect the impact of set-up optimization to be even larger if Factory Planning is being set-up for more machines and a larger set of items. The importance of implementing set-up optimization is not just to reduce the set-up and thus the lead time, but also to represent reality more accurately. From this experiment in which we include set-up optimization we discover that Factory Planning does not detect coincidental set-up time reduction. This inaccuracy of Factory Planning could cause disturbances on the shop floor and sequentially lead to reduced efficiency. It is however important to get reliable data on the set-up times to enhance the accuracy of the schedules in terms of realistically representation the situation at the shop floor.

5 Scheduling Algorithm

From Chapter 4 we conclude that Factory Planning is partly a black box. To gain more knowledge on how to improve the production schedules and what the impact is of including the scheduling extensions we create a scheduling algorithm and also analyze a number of alternative scheduling algorithms.

This chapter answers part b of Sub Question 3: *“What are potential scheduling improvements for PM?”*. Sub Question 3b is: *“What should a new and optimal scheduling process for PM look like?”*. We also address Sub Question 4 in relation to these new scheduling processes: *“What is the impact of the scheduling improvements for PM?”.* We start off in Section 5.1 with the formulation of the scheduling problem. In Section 5.2 we describe the proposed scheduling algorithm. Section 5.3 describes the approach we use to review the performance of the scheduling algorithm. Section 5.4 represents the results of applying the scheduling algorithm to the production schedules. Section 5.5 summarizes the main results from this chapter.

5.1 Problem Formulation

The problem at hand is as follows. Each production department at PM has various machines. The products ordered at PM by external customers result in the various jobs consisting of one or multiple operations that need to be performed by the machines. Some operations need to be executed by one specific machine, other operations can be executed by multiple machines. For a machine to execute the operation, a worker needs to be present at the machine. A machine can either be semi-automatic or non-automatic. In case of a semi-automatic machine a worker only needs to be present during set-up time. For non-automatic machines a worker is required during the complete processing time.

In the scheduling process, operations and workers need to be assigned to machines such that the OTDP is maximized. In order to achieve an optimized OTDP and at the same time make the schedule robust against unforeseen events that can cause disturbances on the shop floor, recall from Chapter 3, the objective function is minimizing the lateness. This measure encounters earliness as negative lateness. By minimizing the lateness, the scheduling algorithm is more prone to accept schedules that contain buffer time which means a job is finished earlier than its due date. This might seem undesirable, because being finished before the customer actually wants to receive the product, makes that the finished product must be kept in stock at PM. This increases stock costs and reduces liquidity of the company, since money is fixed in the finished product and can therefore not be used for other investments. On the other hand however, being finished before the due date creates room in terms of machine capacity and time to be able to deal with unforeseen disturbances such as machine failures or rush orders. By anticipating on these disturbances it is more likely to eventually deliver on time and hence improve the OTDP.

To include the DRC aspect of the job shop at PM, we take into account the labor flexibility and the worker assignment. As for the labor flexibility we do not consider the possibility of training workers to a certain skill level. We include the flexibility of the workers as it currently is at PM. By labor flexibility we mean the number of different machines a worker can attend. In modeling this scheduling problem, for each machine we specify a group of workers that can attend this machine. Depending on the skill level of a worker, a worker can be assigned to one or multiple machines, however not at the same time. A worker that is capable of operating on two different machines has a higher flexibility than a worker that is capable of operating on only one machine. A worker can be assigned to one machine at a time. Whether the worker is only assigned during set-up or during the complete processing time including set-up depends on whether the machine is semi-automatic or not. The set-up time required for a certain operation, depends on the preceding operation. The set-up times are based on the currently known set-up times as defined in Glovia.

Constraints and Assumptions

To achieve a feasible schedule, the following constraints must be met:

1. only one operation can be processed at a machine at a time,
2. a worker can be assigned to one machine at a time,
3. an operation can only be scheduled at a machine if a worker is assigned to the same machine during set-up and processing time or only set-up time for not automatic and semi-automatic machines respectively,
4. the operations of one job must be processed sequentially, i.e. an operation may not start until all its predecessor operations are finished,
5. once an operation is scheduled on a machine, this operation may not be preempted and must be finished on that machine,
6. a worker may only be assigned to a machine he/she is capable of attending that machine,
7. an operation must be assigned to a machine that can execute the required process for this operation,
8. working overtime is not allowed, so when an operation cannot be completed within the regular available hours, the processing of the operation continues the next day.

Besides these constraints, we make the following assumptions. When the scheduling process starts, all machines and workers are available and all jobs are released. This is in line with how Factory Planning approaches the scheduling process. There is however one exception. When operations are still being processed when scheduling is again initiated, Factory Planning pins these operations, such that this part of the schedule remains intact according to how these operations are scheduled in the previous schedule. For the purpose of this research we neglect this exception. We assume all workers have the same experience and thus all work at the same speed. This is not realistic. So when applied to practice, one should consider different working speeds of workers. Also we assume all workers are fulltime employed and are present 5 days a week, 9 hours per day, of which 8 hours are working hours. So the workforce is homogeneous. This is also not realistic, but for the purpose of this research we decide not to increase complexity at this point. Including heterogeneity of the workforce is an interesting subject for further research and must also be considered when applied to practice. For the processing times, we stick to the processing times that are known and available in the ERP system Glovia. We assume the processing times are deterministic. This is how PM currently approaches the processing times. For the actual processing time after the set-up time, this is reasonable for at least the Computer Numerical Controlled (CNC) machines, because once the program for the machine is prepared, the machine runs exactly the same program and thus, if no unforeseen events occur, the duration does not show variation. For set-up activities and machines that require manual activities processing times may vary more. We neglect transportation time. This is also the case in the Glovia data. Finally, we assume that loading and unloading time is included in the processing time. This assumption is also currently applied at PM.

Objective Function

The objective we use for the scheduling algorithm for PM is to minimize the lateness. To calculate the lateness, we subtract the due date from the finished date of each job and add them up. A job is finished when its last operation is finished. How long it takes to finish all operations of a job depends on the available machine and worker capacity. A workday at PM consists of 8 regular workhours. During these hours all workers are present. Per machine the capacity used to schedule jobs varies. Sometimes the machine capacity is lower than 8 hours per day. This does not mean the machine is not active for 8 hours a day, but this lower capacity is used in Factory Planning to be able to execute the schedule, that Factory Planning creates, in practice. So the available capacity is adjusted according to experience from practice. For many machines at PM the performance is currently set to 75% in Factory Planning. This means that only 75% of the capacity may be used to schedule. Not using 100% performance for all machines has the same reason as for the decreased number of hours available per day. There are also some machines that have a capacity of more than 8 hours per day. This holds of course for machines that are semi-automatic and thus can run partially without requiring a worker. So, these machines are not restricted to the 8 hour workday.

When we also include workers as a resource constraint we must think about how to take into account the machine and worker capacities in cases where they are not the same. Since the reason for a lower machine capacity is that in practice it turns out that operations or secondary activities such as loading takes longer than expected, it makes sense that also the worker is required longer than just the scheduled processing times of the operations. The capacity implicitly represents the speed of a machine, because a lower capacity means that less hours can be used to schedule operations, but the machine will be utilized the whole day, so 8 hours. So, in case a machine has a capacity lower than 8 hours per day, we take this capacity as a pacer for the worker. Even though the worker has a capacity of 8 hour per day, when assigned to an operation that is processed on a machine that has a capacity lower than 8 hours per day, the capacity of the worker reduces to the capacity of the machine. For example if a worker is assigned to an operation of 6 hours that is processed on a machine that has 6 hours capacity available per day, this machine requires the whole working day, which is 8 hours. So, in this example the worker is busy the whole day even though this operation requires 6 hours if this operation is scheduled on this machine.

Another case is that a machine has more than 8 hours available per day. If we include workers as a resource constraint, but do not include semi-automatic machines, we assume the worker is required during regular working hours, so 8 hours per day. For the machine capacity after these 8 regular hours we assume no worker needs to be scheduled. This holds for operations that have a duration of more than 8 hours, after these 8 hours, the processing of the operation continues by the machine while the worker is disregarded. This might seem as if semi-automatic machines are included, even though we consider the case that we do not include semi-automatic machines. The case in which we consider semi-automatic machines is however different, because in that case we only require the worker during set-up. Also this is how the capacities are used in Factory Planning, so we decide to stick to these capacities. To represent this we again consider the machine capacity as a pacer for the worker capacity. For the same case, when operations take less than 8 hours, the worker is just fully occupied according to the worker capacity. In the case for which we do consider semi-automatic machines, for the operations scheduled on machines that have a capacity of more than 8 hours per day and thus are semi-automatic, the worker assigned to this operation has a capacity of 8 hours, because the worker is only present during set-up.

To conclude, we want to minimize the lateness, this is the objective in this scheduling problem. The lateness depends on the finished times of the last operation of each job. How fast all operations can be processed by the machines and workers depends on the capacities of the machines and the workers. Whether the machine capacity is the pacer for the worker capacity or the 8-hour workday depends on what case we consider.

5.2 Proposed Algorithm

Based on the literature that Chapter 3 describes, we decide to base our algorithm on the hybrid metaheuristic that Wu et al. (2018) describe, recall Chapter 3. Wu et al. (2018) combine the GA with a VNS that enhances the local search ability of the GA. This research shows good results for the proposed GA-VNS which Wu et al. (2018) apply to the Dual Resource Constrained Flexible Job-shop Scheduling Problem with Learning Effect (DRCFJSP-LE). The DRCFJSP part of the problem they address, complies with the situation we consider at PM. Since this is recent literature on our topic and the results are positive for the case in the research, we decide to take this algorithm as starting point. To implement the scheduling algorithm we make however some adjustments. First of all, we do not include the learning effect of the workers. This is not the main focus of this research, but might be interesting to implement in further research on the case of PM. Second, we adjust the objective of the optimization process. Wu et al. (2018) aim to minimize the makespan. We are, however interested in minimizing the lateness, so this is our objective function instead of minimizing the makespan. Third, we include set-up optimization. We do not strictly optimize the set-up time, since this is not our main aim, but we include set-up optimization in the construction of the initial schedule. Also, because the set-up optimization positively influences the objective value the optimization algorithm is more prone to accept schedules where succeeding operations have the same set-up parameter. Choosing the combination of operations that minimizes the set-up time reduces the required machine and worker capacity. Increased capacity is likely to reduce the lead time of the orders, which in turn increases the probability that sales orders can be shipped on time. Fourth, we include semi-automatic machines. Wu et al. (2018) assume that during the total processing time of an operation a worker needs to be assigned to the machine. At PM there are machines that run automatically after set-up. So, for semi-automatic machines, after set-up the worker is available again to be assigned to another machine.

The GA-VNS algorithm consists roughly of five steps:

1. Population initialization
2. Selection of individuals for the next generation
3. Crossover
4. Mutation
5. Local search: VNS

After Step 5 a new population has been generated. Steps 2 through 5 are repeated until the predefined number of generations are run through. The idea of GAs is to iteratively create new populations of new individuals which all represent a schedule. New individuals are generated by crossover and mutation. Individuals are evaluated based on their fitness. The fitness corresponds to the objective value. By preferring individuals with a better fitness value in the selection step, each iteration the population contains individuals representing a schedule of improved quality comparing to the previous population. Figure 22 displays the flow chart of the GA-VNS.

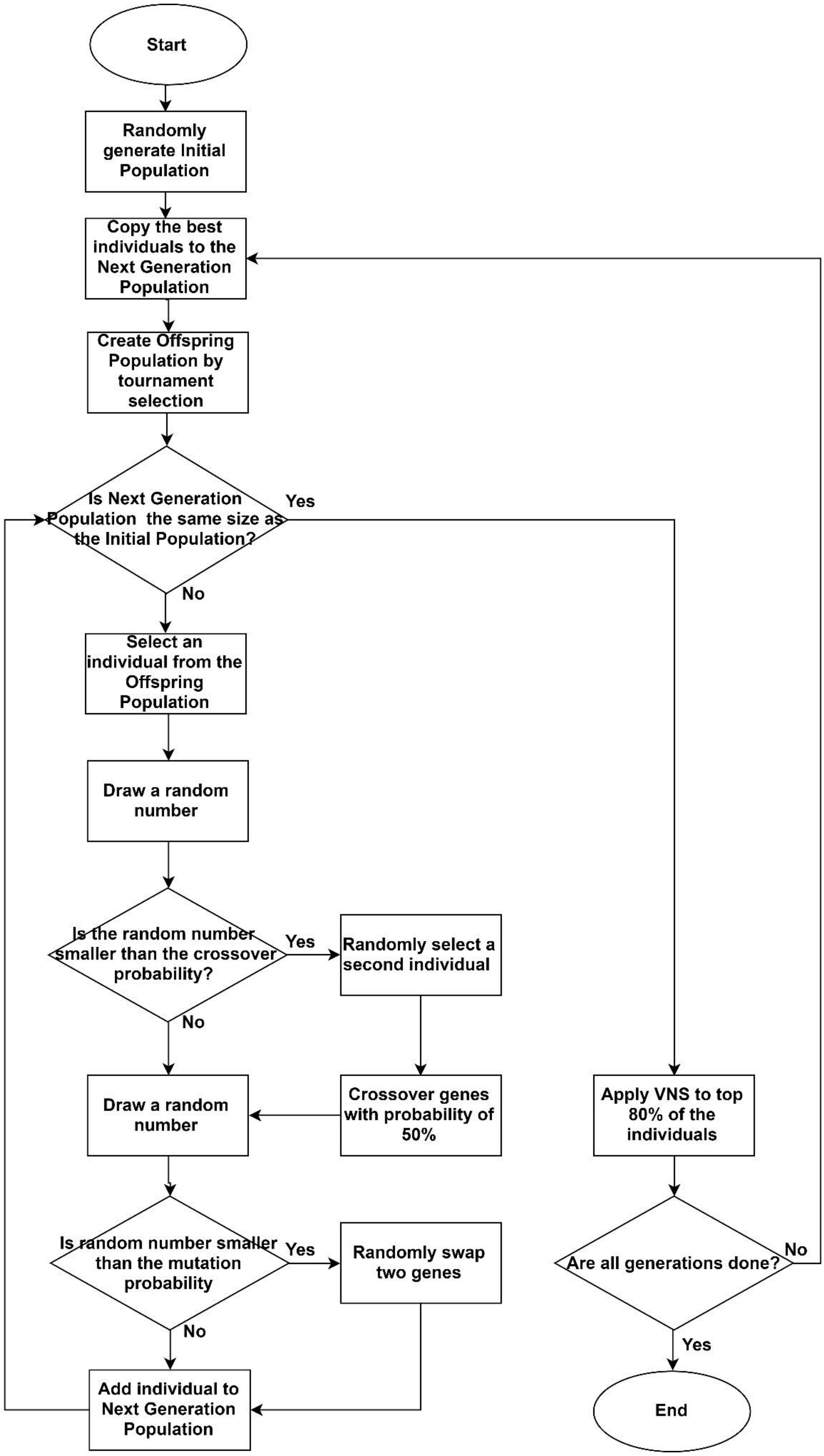

Start
Randomly generate Initial Population
Copy the best individuals to the Next Generation Population
Create Offspring Population by tournament selection
Is Next Generation Population the same size as the Initial Population?
Yes
No
Select an individual from the Offspring Population
Draw a random number
Is the random number smaller than the crossover probability?
Yes
No
Randomly select a second individual
Crossover genes with probability of 50%
Draw a random number
Is random number smaller than the mutation probability
Yes
No
Randomly swap two genes
Add individual to Next Generation Population
Apply VNS to top 80% of the individuals
Are all generations done?
No
Yes
End

Figure 22: Flow Chart GA-VNS

Encoding and Decoding

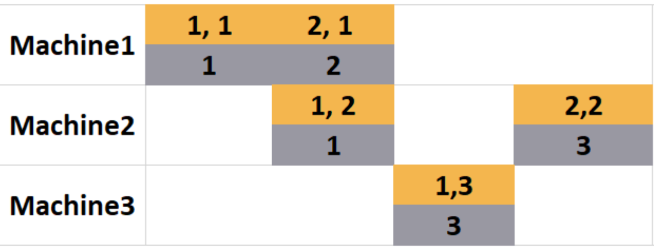
To represent a schedule in the algorithm we need to encode the schedule in a way that at the end of the algorithm the schedule can be decoded again. An encoded schedule, which we refer to as an individual, consists of three chromosomes, see Figure 23. The first chromosome is the operation chromosome and represents the sequence in which the operations are scheduled. The second chromosome represents on which machine this operation needs to be performed. The third chromosome represents the worker that is assigned to this operation and machine. Each chromosome consists of the same number of genes. A gene is a position on the chromosome on which an operation can be scheduled. The number of genes corresponds to the total number of operations that need to be scheduled.

Each individual, which consists of three chromosomes, represents an encoded schedule. No times are included in this representation. In the decoding process the start and end times of the operations and the times at which the machines and workers are available again can be derived from the processing times of each operation and the sequence of the operations. In the decoding process we start at the first gene. The operation at this gene is scheduled first on the machine that we can read from the machine chromosome. Also, the worker that we can read from the worker gene gets assigned to this operation on this machine. Each next step in the decoding process, the next gene is considered. The schedule starts at time 0. So all machines, jobs and workers are available at time 0. We refer to the time at which a job, machine or worker is available as the release time. Each next decoding step the release time of the job, machine and worker involved in the gene under consideration gets updated.

1,1	2,1	1,2	1,3	2,2	Operation Chromosome
1	1	2	3	2	Machine Chromosome
1	2	1	3	3	Worker Chromosome
Gene1	Gene2	Gene3	Gene4	Gene5	

Figure 23: Schedule Representation

We use an example to explain how we calculate the start and finished times of the jobs. Suppose we have 2 jobs, of which Job 1 consists of 3 operations and Job 2 consists of 2 operations. To represent a schedule of this problem we need 5 genes, because that is the total number of operations of all jobs that need to be scheduled. We have 3 machines and 3 workers. Figure 23 shows a possible encoded schedule for the example problem. Such an encoded schedule is the result of running the GA-VNS. For decoding we have to take into account the release times of the job, machine and worker. The latest release time of these three is the start time of the next operation. For the first gene this is no problem, since the jobs, machines and workers are all released at time 0. So, Operation 1 of Job 1 (1,1) is assigned to Machine 1 and Worker 1 will attend this machine at time 0. Suppose the processing time of Operation 1,1 is 4 hours, the new release time of Job 1, Machine 1 and Worker 1 equal the start time at which this gene is scheduled plus the processing time of Operation 1,1. In this case this is 0 + 4 = 4. For Gene 2 we again check the release times. For Gene 2 the first operation of Job 2 is scheduled, so the release time of the job is 0. The machine has release time 4, because first Operation 1,1 is processed on this machine. Worker 2 has release time 0. The latest release time is the start time for this operation, so Operation 2,1 and Worker 2 are assigned to Machine 1 at time 4. The release times of the operation, machine and worker is again the start time plus the processing time of Operation 2,1. Suppose the processing time of Operation 2,1 is 1 hour, the release times are 4 + 1 = 5. The remaining of the encoded schedule is decoded in the same way. Figure 24 displays the completely decoded schedule of this example. The top number in yellow is the operation number. So 1,1 is Operation 1 of Job 1. The bottom number in grey represents the worker number. The length of the colored cells in Figure 24 does not represent the duration. For the actual schedule one would want to include the finish/start times of each operation. For simplicity of the figure we left that out, but can be included for example at the bottom x-axis of the figure.



Machine1	**1, 1** **1**	**2, 1** **2**		
Machine2		**1, 2** **1**		**2,2** **3**
Machine3			**1,3** **3**	

Figure 24: Decoded Schedule

Worker Constraint

Since the worker is currently not considered as a constraint in the scheduling process, adding this constraint might result in schedules that seem to be worse. For example the makespan might increase, because a constraint is added. Including this constraint however, will more realistically represent reality. This results in a schedule that is more likely to be realized and thus we expect that including this constraint will increase the OTDP at PM.

To include both machines and workers as a resource constraint we need two pieces of information when we want to schedule an operation. We need to know which machines are capable of executing this operation. We also need to know which workers are capable of attending each machine. To structure this information we introduce two terms. The first is “Machine-group”. A machine-group is a set of machines. A machine-group consists of at least one machine. A certain machine can be assigned to one or multiple machine-groups. We use the machine-groups to define for each operation what set of machines is capable of executing the operation. So, each operation is related to one machine-group. Operations that require a machine from the same set of machines are related to the same machine-group. The second term we introduce is “Machine-cluster”. A machine-cluster is a set of workers. A machine-cluster consists of one or multiple workers. A certain worker can be assigned to one or multiple machine-clusters. For each machine we define to which machine-cluster they are related. Each machine is related to one machine-cluster. This machine-cluster defines which workers can be assigned to the machine. Different machines can be related to the same machine-cluster if they happen to require a worker from the same set of workers. So, a machine-group is a set of machines related to an operation and a machine-cluster is a set of workers related to a machine.

When we need to schedule a certain operation of a job we first need to select a machine from the machine-group of this operation. Then we need to select a worker that belongs to the machine-cluster of this machine. To illustrate this part of the scheduling process, we present an example, see Table 3 to Table 5. Say we need to schedule Operation 1 of Job 2. This operation is related to Machine-group 3. To pick a machine, all machines that belong to Machine-group 3 are an option. Table 4 shows that Machine 3, 5 and 6 belong to Machine-group 3. So, to schedule Operation 1, we can select either Machine 3, 5 or 6. If we select Machine 5, we need to pick one of the workers from Machine-cluster 4, because this machine is related to this machine-cluster. So, we can either select Worker 5, 6 or 7 to be assigned to this machine. If we pick Worker 6, Operation 1 is scheduled on Machine 5 and Worker 6 will attend this machine.

Job1	Operation1	>> Machine- group1
	Operation2	>> Machine-group1
	Operation3	>> Machine-group2
Job2	Operation1	>> **Machine-group3**
	Operation2	>> Machine-group1
Job3	Operation1	>> Machine-group4
	Operation2	>> Machine-group1
	Operation3	>> Machine-group2
	Operation4	>> Machine-group5

Table 3: Operations

Machine-group1	Machine1	>> Machine-cluster1
	Machine2	>> Machine-cluster1
Machine-group2	Machine3	>> Machine-cluster2
	Machine4	>> Machine-cluster3
Machine-group3	Machine3	>> Machine-cluster2
	Machine5	>> **Machine-cluster4**
	Machine6	>> Machine-cluster2
Machine-group4	Machine1	>> Machine-cluster1
	Machine4	>> Machine-cluster3
Machine-group5	Machine6	>> Machine-cluster2

Table 4: Machine-groups

Machine-cluster1	Worker1
	Worker2
	Worker3
	Worker4
Machine-cluster2	Worker3
	Worker5
Machine-cluster3	Worker5
	Worker6
Machine-cluster4	Worker5
	Worker6
	Worker7

Table 5: Machine-clusters

Algorithm Parameters

To apply the GA-VNS, we need to define several parameters for the algorithm. These parameters are the number of generations, a population size, selection operator, crossover operator, mutation operator, crossover probability, mutation probability, neighborhood structures for the VNS and number of VNS iterations. The value of each of these parameters influence the solution quality and the required computing time.

The number of generations represent the number of iterations for the GA-VNS algorithm. An additional generation severely increases the computation time, since the whole algorithm including the VNS local search is repeated. Increasing the number of generations does, however provide the opportunity to improve the solution since for each new generation the population of the previous generation is used as input.

A population is the collection of all the individuals of a generation. Each individual represents a feasible schedule for the problem under consideration. The population size both influences the computation time and the quality of the final solution. A small population size might result in inferior results, but as the population size increases the computation time increases, which is undesirable. The optimal population size depends on the problem type (Roeva, Fidanova, & Paprzycki, 2013). Each next generation the population contains the offspring of the individuals in the population of the previous generation. To determine which offspring individuals make it to the next generation we apply a selection operator. We decide to apply both elitist preservation and tournament selection to create a new population. These two selection operators seem to work well in a similar problem (Wu, Li, Gou, & Xu, 2018).

The selection of the next generation population starts off with elitist preservation. This selection operator preserves the best individuals. The elite reserve ratio determines what percentage of the population size is copied directly to the new population without crossover and mutation. A high value for the elite reserve ratio increases the convergence speed of the GA, because the best solution is remained. At the same time however, a high elite reserve ratio decreases the diversity of the population which might result in a solution of inferior quality. We follow the value for the elite reserve ratio of Wu et al. (2018) who propose an elite reserve ratio of 0.05. So, the top 5% of the individuals of the population are directly copied to the new population without crossover and mutation.

To create a new population of the same size as the initial population, we use tournament selection in addition to the elitist preservation. For this method the GA randomly selects a predefined number of unique individuals from the parent population. From this set, the individual with the highest fitness value ‘wins’ the tournament and is selected and placed into the offspring population, see Figure 25. This process continues until the offspring population consists of 95% of the population size, such that the next generation population is the same size as the initial generation. The number of individuals that is selected for each tournament is the tournament size. For each tournament all individuals of the initial population have the same probability of being selected for the tournament. The larger the tournament size, the smaller the variability of the next generation population. If the tournament size is large, the individuals with inferior fitness values have a small chance of winning the tournament. To illustrate, if the tournament size is 3, the two individuals that have the worst fitness value will never make it to the offspring population, because there is always a third individual that has a fitness value that is higher than the fitness of these two individuals. The only exception is in case the fitness of the third individual equals the fitness of one of the two worst fitness values. The number of individuals that do not make it to the next generation is the tournament size minus 1, bearing in mind the exception of equal fitness values. For this reason we want the tournament size to be small. We choose 3 as the tournament size.

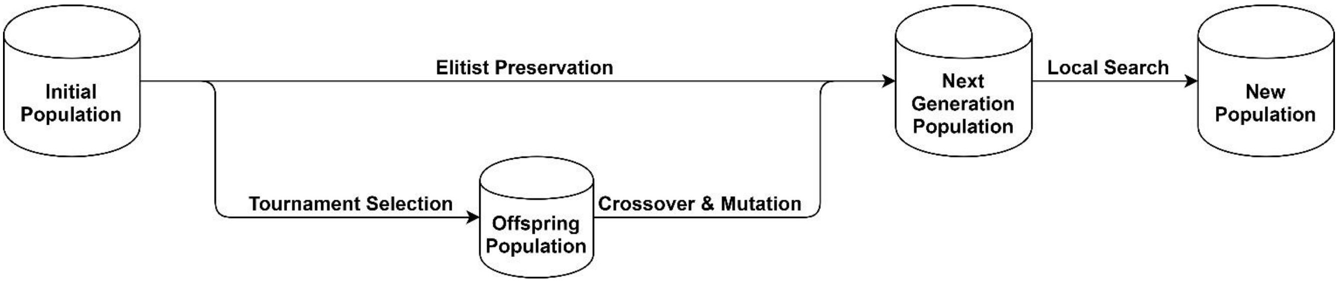


Figure 25: Genetic Algorithm Process

Crossover and mutation are important operators within the GA to generate new solutions. The crossover operator enhances the converging behavior of the GA. The mutation operator, which randomly introduces changes into the chromosomes, serves to retain diversity. Diversity is important for escaping from local optima which could result in premature convergence (Konak, Coit, & Smith, 2006). The crossover probability determines how likely it is that a crossover between two individuals takes place. The mutation probability determines the likelihood of an individual undergoing mutation. For the crossover probability we initially choose a value larger than 0.5. We can afford to widely explore the search space, because by applying elitist preservation the best solution does not get destroyed. The value used for the mutation probability is generally low, between 0.005 and 0.1 (Sanches, Rocha, Castoldi, Morandin, & Kato, 2015) (Wu, Li, Gou, & Xu, 2018) (Calzarossa, Vedova, Massari, Nebbione, & Tessera, 2019). It is however useful to consider higher values for the mutation probability if the population size is small. If the population size is small it might happen that no individual undergoes mutation. So the purpose of mutation, keeping the population diverse, is lost.

The VNS proposed by Wu et al. (2018) uses three neighborhood structures. The first is "exchange" and focuses on the operation chromosome. For exchange, randomly two different operations on a chromosome are selected and exchanged. The second structure is "replace" and focuses on the machine chromosome. For replace randomly a gene is selected and for this operation another machine from the work center to which this operation is related is selected. A worker is selected that is assigned to the machine-cluster of this machine. The third structure, "change", focuses on the worker chromosome. For change, a machine is randomly selected for which the worker assigned is changed. We also need to determine the number of VNS iterations. Next to the number of iterations it is important to note that VNS is not applied to each individual. The best individual are improved by VNS. The worst individuals remain unchanged to account for diversity in population.

Parameter Tuning

To tune the parameter values such that the performance of the algorithm increases, we run several experiments. The number of generations, population size, crossover probability, mutation probability and the number of VNS iterations are the parameters that we expect to have most impact on the solution quality. This is based on what seems common practice in literature. For the elite reserve ratio and the tournament size we use the values proposed by Wu et al. (2018), respectively 0.05 and 3. To determine the values for the other parameters we execute several experiments to see what the impact is of adjusting these parameter values. We start off with relative low values for each parameter. We refer to this as the default setting. Table 6 displays the default settings for these parameters.

We run de GA-VNS twice 50 times. For each of these 50 runs one parameter value changes. The first time we execute the 50 runs we change the parameters by a small increment. For the second time, we use a large increment, see Table 6. We do not consider a distinction between a small and large increment for the population size and number of VNS iterations. This is based on the results of preliminary experiments. From these experiments we observe that the other three parameters do not affect the performance much. We expect by applying a larger increment for these parameters the effect will be larger. This has to do with our experimental design. If the change of a parameter yields an improvement of the fitness value, this change gets fixed. If a parameter change does not improve the algorithm performance, this parameter is reset and another parameter is changed. If a parameter value does improve the performance, this parameter is increased again, until this does not yield an improved performance anymore. If the increment is too small to have any affect, the value of this parameter will never change even though a larger value might be better.

Parameter:	Default Setting:	Small Increment:	Large Increment:
Number of Generations	2	1	2
Population Size	5	5	5
Crossover Probability	0.5	0.1	0.2
Mutation Probability	0.005	0.001	0.002
Number of VNS iterations	2	1	1

Table 6: Default Algorithm Parameter Settings

Since the algorithm is based on stochastic values, for each run we execute multiple replications. To determine whether a change in a parameter value improves the algorithm performance we compare the average fitness value of all the replications and compare this to the average fitness value of the previous parameter setting.

To improve these parameter settings, we take a small instance to speed up the process. To see if any parameter is dependent on the size of the instance, we also take a medium and large sized instance. The medium and large instance are still relative small instances. For the parameter values we run 4 experiments. The first experiment is a small instance that consists of 5 jobs. The second experiment is also a small instance, but consist of 5 different jobs than the first experiment. The third experiment is a medium sized instance. This instance contains 10 jobs. The fourth experiment is a large instance and consist of 15 jobs.

We do not see any pattern in the resulting GA-VNS parameters for the various problem sizes. So, we assume that the size of the problem does not significantly influence the optimal parameter settings. From the experiments we observe that the values of the crossover and mutation probabilities do not increase. Also when we disable the increment of generations and focus on the probabilities, these do not change. Also the number of VNS iterations do not increase. Table 7 shows the parameter values we propose based on the executed experiments.

Parameter:	Proposed Value:
Number of Generations	6
Population Size	30
Crossover Probability	0.5
Mutation Probability	0.005
Number of VNS iterations	2

Table 7: Proposed Parameter Values

Now we know the parameters we want to review the value of the mutation probability, because this value is usually kept very low. In our case however for a population size of 30 and 6 generations, a mutation probability of 0.005 causes on average, when repeated many times, 0.15 individuals in each generation to undergo mutation. So, over 6 generations on average not even 1 individual mutates (30*0.005*6 = 0.9). To increase the impact of the mutation we increase the mutation probability to 0.035. In this case, on average, when repeated many times, 1.05 individuals each generation undergo mutation (30*0.035 = 1.05). This is still a low number of individuals that undergo mutation, but compared to the population size and keeping in mind that literature proposes a low mutation probability, we think this is a good value for the mutation probability.

Alternative Scheduling Algorithms

Next to this GA-VNS, which is our main focus, we also analyze several other scheduling algorithms to put the scheduling results in perspective and for validation purposes. Since the GA-VNS is a quite extensive algorithm we are interested in comparing this algorithm to two more straightforward local search algorithms. The first we want to address is Simulated Annealing (SA). This is a well-known meta-heuristic and widely applied to various combinatorial optimization problems, among which the JSP (Yazdani, Zandieh, Tavakkoli-Moghaddam, & Jolai, 2015). Yazdani et al. (2015) apply SA to the DRCFJSP which shows promising results.

For the SA we use the same schedule representation as for the GA-VNS, see Figure 23. As neighborhood structure we use swapping operations. Two genes get selected randomly, the whole gene gets swapped. This means that once a machine is assigned to an operation this machine remains assigned to this operation. The same holds for the workers. For each iteration the objective value of a neighbor solution is evaluated and compared to the original schedule. If the objective value of the neighbor solution is better than or equal to the original schedule, this new schedule is accepted and becomes the new original schedule. Whenever the neighbor schedule is however inferior to the original schedule, the new schedule is selected only with a certain probability which is dependent on the value of the temperature. Since the temperature decreases over time, the probability of accepting an inferior schedule decreases as well. This makes SA both a random search and a local search method.

To run this algorithm we need to define the cooling scheduling. A cooling schedule consists of various parameters that need to be defined. These parameters are as follows. The initial temperature is the initial value for the algorithm. Every iteration the temperature decreases. To calculate the new decreased temperature, we multiply the current temperature by a value alpha, which must be smaller than 1. To slowly decrease the temperature the value of alpha should be close to 1. The larger the value of alpha the slower the temperature decreases. The algorithm stops when the temperature falls below a certain defined stopping temperature C_{Stop}. This value is somewhere close to 0 for example 0.001 (Dowsland & Thompson, 2012). For each temperature value there is a predefined number of iterations for each temperature, the length of the Markov chain. After all iterations are done for one temperature value, the temperature gets decreased by the factor alpha. The algorithm stops when the stopping temperature is reached. In the decision on what Markov chain length to use, lies the tradeoff between computation time and solution quality. A too low value for the Markov chain length makes that a suboptimal solution is reached. A too high value for the Markov chain length results in excessive computation time while the improvement in the solution is only marginal. Some authors advice to increase the Markov chain length for each new temperature, because the probability of accepting inferior schedules decreases.

Since we additionally set up the SA and this is not our main focus in this research, we do not apply an extensive parameter tuning procedure. Also, to be able to run all the experiments in the time we have, we propose a rather small Markov chain length. Table 8 shows our proposed values for the SA parameters.

Parameter:	Proposed Value:
Starting Temperature	20
Alpha	0.9
Markov Chain Length	10
Stopping Temperature	0.001

Table 8: Simulated Annealing Proposed Parameter Values

From the results of a couple of initial experiments on both the SA and GA-VNS we observe that SA does not perform a lot worse than GA-VNS. Because of this observation we replace the VNS by a simple local search procedure. So, instead of applying various neighborhood structures we only apply the operation swap structure from SA. Because we expect this simpler local search procedure to be faster, we decide to increase the number of iterations to 10. We observe that this adjustment in the GA-VNS shows more promising results for the scheduling problem.

A possible cause for this can be the number of possible neighbors for each of the three different neighborhood structures of the VNS. The first neighborhood structure of the VNS is “exchange”. This works the same as the swap neighborhood structure we apply in SA. This search space is very large because we can swap any combination of operations. The second structure is “replace”. For this neighborhood structure we replace the machine that is scheduled for a certain operation. This search space is much smaller, because not all operations have a machine-group that consists of more than one machine. If the machine-group consists of only one machine, the replace structure cannot yield any change for this operation. The third structure is “change”. This structure changes the worker that is assigned to a certain machine. For this neighborhood structure we have the same situation as for the replace structure. Only a worker that belongs to the machine-cluster of a machine can be assigned to this machine. For some machines the related machine-cluster is small or only contains one worker. So, also in this case the neighborhood structure fails to find a feasible neighbor.

The VNS is a more extensive local search procedure and thus requires more computation time than the NS local search procedure. If we only consider the exchange, i.e. swap, neighborhood structure, the computation time is less, while at the same time we remain a rather good neighborhood structure.

Because we replace the Variable Neighborhood Search (VNS) by a simple Neighborhood Search (NS), we call the algorithm that that only considers operation swap as neighborhood structure GA-NS. For the GA-NS we do not observe any outstanding results in the parameter tuning experiments. We decide to keep the input parameter, with the exception of the number of local search iterations, as defined for the GA-VNS, recall Table 7.

To summarize, we expect the exchange neighborhood structure to have most impact on the objective function, so we decide to additionally analyze the GA-NS. For this algorithm we do however increase the number of local search iterations to be able to achieve a better performance. By this change we expect the computation time not to increase significantly.

To get an idea of the performance of these algorithms we additionally apply a simple local search heuristic. Steepest Descent is a local search algorithm that aims to improve a constructed schedule.

For a given schedule Steepest Descent reviews all its neighbor schedules. The algorithm searches for the best neighbor and accepts it if it improves the objective value of the given schedule. If a neighbor is accepted all neighbors of this schedule are reviewed. This process continues until no improvement can be found in the neighborhood of a schedule. This scheduling procedure is in literature also addressed as the Steepest Hill Climbing technique (Di Gaspero, 2003). Literature does not propose a neighborhood structure for this specific scheduling problem, DRCFJSP, in combination with our objective value. We define the neighborhood structure as swapping two genes. So a neighbor is the resulting schedule from swapping two genes of the original schedule. To reduce the number of neighbors, which increases the computation time, we decide to only swap adjacent genes. For this algorithm we do not need to define any parameters. The local search stops when none of the neighbors yield an improvement.

Finally we also consider the combination of GA-VNS and SD (GA-VNS+SD) and GA-NS and SD (GA-NS+SD). For these two algorithms we apply SD when the GA-VNS respectively GA-NS is finished. We also include these algorithms, because we are interested in whether applying SD afterwards the extensive GAs yields any improvement.

Validation

To be able to draw valid conclusions from our scheduling algorithm, we check the validity of our scheduling algorithm. It is important that this algorithm applies the assumptions that we made for the scheduling problem at hand, otherwise the conclusions might not be applicable to the problem at hand in this research. During the process of programming the algorithm we extensively debug the code to detect mistakes. Also we run the algorithm for a very small problem instance with only a few operations to schedule. While executing the code step by step, we manually do the calculations to check whether the algorithm does what we want it to do. This turns out to be all correct.

Since the algorithm is not the same as for Factory Planning we can unfortunately not compare the scheduling result from our algorithm to the scheduling result from Factory Planning. We can however compare our results to the results of another algorithm. We do however critically analyze the results of the various algorithms and see if the results are as expected.

Additionally we program a decoding procedure in Excel – VBA. This decoding uses as input the output of the GA scheduling algorithm. By going through the created schedule step by step, this decoding procedure visually represents the schedule in a Gantt chart and also calculates the lateness. Although we already check the feasibility of a schedule during the GA, again we can check the feasibility when the schedule is decoded. From this validation process we can conclude that our scheduling algorithm is valid to execute experiments and draw conclusions for our research.

5.3 Application of the Scheduling Algorithm

To evaluate the performance of the scheduling algorithms we develop the scheduling algorithm using Visual Basic. Appendix C: Pseudo Code shows the pseudocode of the GA-VNS scheduling algorithm. We implement this algorithm in Visual Studio to analyze the performance of the scheduling algorithm. We consider the four scheduling algorithms GA-VNS, GA-NS, SA and SD, we discuss in Section 5.2 of this chapter.

Ideally we want to compare the results of the algorithms to the results of Factory Planning. In the process of modelling the situation at PM we come to the conclusion this is way too complex. First of all Factory Planning considers the production, assembly and outsourcing processes. The production processes require both a machine and a worker. The processing times are expressed as number of hours and weekend days are excluded. For the outsourcing processes the duration is expressed as the number of days and weekend days are included. For most assembly processes the worker is considered as the machine and this is the only resource constraint that is considered. Since we would want to use, amongst others, the start- and end times of the operations scheduled, these inconsistencies in processing times make it complex to represent the processing times in our algorithms. We solved this problem by introducing dummy operations. We define the operations that are not part of the production processes and thus do not need to be scheduled on the machines and workers of the production departments, as dummy operations. To be able to compare the schedules of our algorithms to Factory Planning we have to include these dummy operations, because the next operation can only start when this preceding operation is finished. So to account for this time the next operation has to wait we include these dummy operations. Whenever a dummy operation is the last operation of a job, we do not need to include this operation in the scheduling process because the precedence constraint is not an issue. In this case we correct for the processing times of these dummy operations by bringing the due date forward. In case of outsourcing operations we convert the durations to hours and correct for the weekend days. In the process of comparing the scheduling results to Factory Planning we however discover that Factory Planning allows overtime. This overtime varies greatly between days and machines. This makes it hard to represent the situation that is the starting point that Factory Planning uses. So, it might be the case that for a certain order set it is not possible to find a feasible schedule in regular production time. Because of these kind of inconsistencies and the limiting experimenting possibilities in the test environment of Factory Planning we are not able to accurately compare the performance of our scheduling algorithms to Factory Planning.

Instead we evaluate the impact of including several scheduling extensions. For this purpose we are interested in four scheduling scenarios. We consider the following scenarios. The first scenario is the base case (Scenario 1). This scenario closely resembles the circumstances under which Factory Planning schedules. In this scenario we do not consider set-up optimization. Also workers are not included as a constraint. Because of the latter, whether machines are semi-automatic or non-automatic is indifferent. The second scenario (Scenario 2) includes set-up optimization. The third scenario (Scenario 3) takes into account workers as a constraint. The fourth scenario (Scenario 4) considers the semi-automatic machines in addition to the worker constraint. These three scheduling extensions to the base case represent the possibilities of enhancing the performance of Factory Planning. It is interesting to investigate the impact of each of these extensions on the schedules created and on the practice on the shop floor. In the following we discuss each of the scenarios into more depth.

Scenarios

The first scenario represents the circumstances under which Factory Planning schedules. This scenario aims to schedule the set of orders, on the production machines, which are the only resource constraints. Set-up optimization is not considered. For the base case we are interested in what algorithm performs best. We also vary the mutation probability based on our considerations for the parameter tuning, recall Section 5.2 of this chapter.

Recall from Chapter 2 the principle of set-up optimization. We include this scheduling extension for the second scenario. In short, set-up optimization aims to combine two operations on the same machine such that the set-up time required for the second operation can be minimized. To include the set-up optimization we need information on the processing times and what set-up time reduction can be achieved for certain combinations of operations. The set-up time reduction depends on the item characteristics.

Just as for the Factory Planning set-up optimization we focus on the machines and products that are most promising to achieve set-up time reduction. We focus on the standard linear bearings. This type of product frequently needs to be produced and by just looking at one characteristic of this product, the set-up time can be reduced to a great extent compared to other products. This characteristic is the roll diameter of the linear bearing. For the set-up optimization in Factory Planning we call such a characteristic a parameter type. Each product that contains this parameter type, has a value for this parameter type "Roll Diameter". Whenever a product with this parameter type is produced on a machine the set-up time can be reduced if the next product scheduled on this machine has the same value for this parameter. For the linear bearings the roll diameter of the current and next product scheduled determine how much set-up time is required for the next product. To get an idea of the performance of the set-up optimization we apply this to machines at the drilling department. Because the set-up times for various products on various machines are not yet determined and it is very time consuming to get all this information on each product, we only consider a small selection of one product type. Also there are many varying product characteristics, so it is very complex to fill the complete set-up matrix. We refer to Chapter 4 for additional explanation on the set-up optimization.

To be able to give relevant advice on the scheduling approach for PM, we ideally model the set-up optimization in the way Factory Planning applies set-up optimization. To do so however we need to model the whole scheduling process of Factory Planning. This is not possible since the information on this externally developed scheduling tool lacks a high level of detail. Therefore we have to think of a way to apply set-up optimization in a similar way Factory Planning applies set-up optimization even though the logic for our scheduling algorithms is totally different from Factory Planning. It is most obvious to apply the set up optimization in the process of generating an initial schedule. In this way we actively search for a good initial schedule in terms of set-up optimization. In the further process of optimizing the schedule we do not consider any additional active set-up optimization. When, however, set-up optimization results in reduced lateness, the algorithm accepts this schedule and thus automatically is more prone to accept schedules where two orders are combined such that they result in reduced set-up times. We do not include the set-up time in the objective value, because solely optimizing based the set-up time is not the main aim. We only want to include set-up optimization if this truly improves the lateness. However, reduced set-up times contribute to shorter lead times, which is a secondary point of interest in this research.

Scenario 3 includes the worker constraint. For this constraint we assume all workers are fulltime. This is however not realistic. Also we do not consider the probability of workers being not present due to illness. Also we do not encounter the experience of workers which probably affects the quality and\or the working speed of the worker. For the purpose of this research we want to see the effect of including workers as a resource constraint, so we do not include these details. These details are however important if this constraint is applied to the scheduling process in practice.

If we encounter workers as an additional resource constraint it is interesting to include semi-automatic machines, Scenario 4. As long as workers are not included in the scheduling process this does not affects the schedules. So for this scenario we also consider workers as a resource constraint in the same way as for Scenario 3.

Experiments

From an order pool of 110 orders we randomly create 10 different instances . This order pool is a large set of orders, which is representative for PM. The number of orders the planner has to schedule per day is usually around 15. To represent a random day at PM, we assume there are 15 orders that need to be scheduled. An order can exist of multiple order lines if the customer orders multiple items. By 15 orders we mean 15 order lines, so for these 15 orders we have to schedule the production process of 15 items. The order pool contains many orders of regular items, mostly linear bearings, but also includes a few orders that occur not that often. For the orders of items that occur multiple times the order quantities differ, based on the experience of the planner. Because the order pool is an enlarged representative set of orders for PM, the smaller order sets, which we randomly pull from the order pool, is also representative for an average order set at PM. Compared amongst each other the smaller order sets might vary. This is however reality for PM, so the varying order sets make the experiments representative for PM. We calculate the required processing times based on the order quantity and the machine run and set-up times from the ERP data. For each experiment we run each of the 10 order sets. We do this to get reliable results on which we can base our conclusions for PM. For the purpose of comparing between the various scheduling algorithms we make sure the initial schedule for each experiment and each order set is the same. We do this by initializing the random number generator, which we use amongst others for creating the initial schedule, with the same seed value. By doing so we know for sure that the improvement in lateness of one experiment compared to another is the result of the scheduling algorithm applied.

Currently, when the 15 orders are scheduled, Factory Planning also reschedules the orders that are already present in the schedule. So in fact the order set to be scheduled is way larger. The number of orders present in Factory Planning can be even over 1000 orders. For the purpose of comparing various scheduling algorithms we do not consider these orders because computation time will be very large. For Scenario 2 this might however give underestimated results, because we define the set-up parameters only for a specified set of items, recall Chapter 4, and the order set we consider, is rather small, so the chance the set-up parameters of two orders match is also very small. Most of our order sets, consisting of 15 orders, do not contain at least two items with the same set-up parameter value. So the effect of set-up optimization will be smaller for these experiments than for reality, because in reality Factory Planning also has the large number of orders that are already scheduled in Factory Planning to be able to match two items. So, to be able to measure the impact of including set-up optimization we consider a large order set of 100 orders. These 100 orders are also present in the 110 order pool. For these experiments we do not run multiple order sets to get representative results, since the order set is large. As opposed to the experiments with the 10 different order sets of 15 orders we do not initialize the random number generator, so we get different initial schedules. To account for this randomness, we run 10 replications for each of the experiments.

We execute two sets of experiments to evaluate the impact of the various scheduling scenarios. Table 9 shows the configurations for each experiment of the first set of experiments. We run each scenario for each algorithm and evaluate different combinations of the scheduling configurations. The purpose of this first set is to evaluate the impact of additional scheduling constraints and extensions on the created schedules.

Table 10 shows the configurations for each experiment of the second set of experiments. For this second set we apply for each experiment a post processing procedure at the end of the algorithm. This post processing procedure evaluates the objective value while including the worker constraint and set-up optimization, even though these two extensions are not part of the experiment. We include these two extensions in the post processing procedure because this realistically represents the circumstances at the shop floor at PM. Even though set-up optimization is not included in the scheduling algorithm, it might occur that coincidentally two items are scheduled sequentially that have the same set-up parameter value. In this case set-up time is reduced, because the workers at the shop floor need less or no time to set up the machine for the second operation. Also for the experiments where workers are not included as a constraint in the optimization process, we apply the post processing to get a realistic schedule, because in practice workers are included in the production process and the next operation can only start when the assigned worker is available again. It does obviously not make sense to apply the post processing to the experiments that already include both set-up optimization and workers. That is why the number of experiments of this second set is smaller than for the first set.

To be able to check whether differences in the lateness values of the various experiments, are statistically significant, we use the unpaired t-test. To calculate the t-value in comparing experiment a to b, we use the following formula (Larsen & Marx, 2012):

$$T = \frac{\overline{x}_a - \overline{x}_b}{\sqrt{\left(\frac{(n_a - 1)s_a^2 + (n_b - 1)s_b^2}{n_a + n_b - 2}\right) * \left(\left(\frac{1}{n_a}\right) + \left(\frac{1}{n_b}\right)\right)}}$$

The notation we use, is:

$\overline{x}_a,\ \overline{x}_b$	= the average lateness for experiment a respectively b.
$n_a,\ n_b$	= the number of replications of experiment a respectively b.
$s_a,\ s_b$	= standard deviation of experiment a respectively b.

Additional notation we use in analysing the results is as follows:

μ_a, μ_b	= mean lateness for experiment a respectively b
$\mu_{AST,\ a}$, $\mu_{AST,\ b}$	= mean AST for experiment a respectively b
H_0	= the null hypothesis
H_1	= the alternative hypothesis

For the level of significance, which we denote by alpha (α), which we distinguish from the alpha we introduce for the cooling parameter of SA, we take 0.05. This is common practice in statistics. The number of degrees of freedom (df) equals

$$df = n_a + n_b - 2$$

The confidence level follows from the value of α (confidence level = 1 – α). For an α of 0.05 we have a confidence level of 0.95, so 95%. This means that the probability of not rejecting the null hypothesis while the null hypothesis is also actually true is 95%. Complementary we can state, for the significance level of 0.05, that the probability is 5% that we reject the null hypothesis while it is actually true. This 5% denotes the probability that the undesirable event occurs of drawing the wrong conclusion. For the t-table we refer to Appendix D: T-Distribution Table.

Before determining the mutation probability of the experiments, we first analyze the first four experiments of the first set to determine what value for the mutation probability we use for the other experiments. For now we are interested in the effect of a mutation probability of 0.005 compared to

0.035. For the GA-VNS the mutation probability of 0.035 does not perform significantly better than a mutation probability of 0.005 with a significance level of 0.05. This is also the case for the GA-NS. When we compare the difference between Experiment 1 and 3 and Experiment 2 and 4 the p-values are respectively 0.40 and 0.49. So, there is no statistical significant difference with significance level of 0.05 between the performance of a mutation probability of 0.005 and 0.035 for both the GA-VNS and the GA-NS. So, we decide to apply the mutation probability of 0.035 for the following experiments.

Experiment Nr.	Scenario	Algorithm	Mutation Probability	Set-up Optimization	Worker Constraint	Semi-automatic Machines	Steepest Descent
1	1	GA-VNS	0.005				
2		GA-NS	0.005				
3		GA-VNS	0.035				
4		GA-NS	0.035				
5		SA	N/A				
6		SD	N/A				
7		GA-VNS	0.035				✓
8		GA-NS	0.035				✓
9	2	GA-VNS	0.035	✓			
10		GA-NS	0.035	✓			
11		SA	N/A	✓			
12		SD	N/A	✓			
13		GA-VNS	0.035	✓			✓
14		GA-NS	0.035	✓			✓
15	3	GA-VNS	0.035		✓		
16		GA-NS	0.035		✓		
17		SA	N/A		✓		
18		SD	N/A		✓		
19		GA-VNS	0.035	✓	✓		
20		GA-NS	0.035	✓	✓		
21		SA	N/A	✓	✓		
22		SD	N/A	✓	✓		
23		GA-VNS	0.035		✓		✓
24		GA-NS	0.035		✓		✓
25		GA-VNS	0.035	✓	✓		✓
26		GA-NS	0.035	✓	✓		✓
27	4	GA-VNS	0.035		✓	✓	
28		GA-NS	0.035		✓	✓	
29		SA	N/A		✓	✓	
30		SD	N/A		✓	✓	
31		GA-VNS	0.035	✓	✓	✓	
32		GA-NS	0.035	✓	✓	✓	
33		SA	N/A	✓	✓	✓	
34		SD	N/A	✓	✓	✓	
35		GA-VNS	0.035		✓	✓	✓
36		GA-NS	0.035		✓	✓	✓
37		GA-VNS	0.035	✓	✓	✓	✓
38		GA-NS	0.035	✓	✓	✓	✓

Table 9: Experiment Configurations Set 1

Experiment Nr.	Scenario	Algorithm	Mutation Probability	Set-up Optimization	Worker Constraint	Semi-automatic Machines	Steepest Descent
1	1	GA-VNS	0.035				
2		GA-NS	0.035				
3		SA	N/A				
4		SD	N/A				
5		GA-VNS	0.035				✓
6		GA-NS	0.035				✓
7	2	GA-VNS	0.035	✓			
8		GA-NS	0.035	✓			
9		SA	N/A	✓			
10		SD	N/A	✓			
11		GA-VNS	0.035	✓			✓
12		GA-NS	0.035	✓			✓
13	3	GA-VNS	0.035		✓		
14		GA-NS	0.035		✓		
15		SA	N/A		✓		
16		SD	N/A		✓		
17		GA-VNS	0.035		✓		✓
18		GA-NS	0.035		✓		✓
19	4	GA-VNS	0.035		✓	✓	
20		GA-NS	0.035		✓	✓	
21		SA	N/A		✓	✓	
22		SD	N/A		✓	✓	
23		GA-VNS	0.035		✓	✓	✓
24		GA-NS	0.035		✓	✓	✓

Table 10: Experiment Configurations Set 2

To distinguish from Set 1 and Set 2 which use the small order sets, we define a third order set. This set consists of 4 experiments that use the large order set for the purpose of getting insight in the effect of set-up optimization. Since the computation time is very high for these experiments we decide to only run the set-up optimization (Scenario 2) for the GA-NS. We do not consider the other algorithms nor the other scenarios because we are interested in the AST which is mainly impacted when we include set-up optimization. We do not have a clear reason for what algorithm to pick for these experiments so we just decide to take the GA-NS, because this is a more advanced algorithm than SA and SD. Also by seeing the intermediate results we think this is a good choice. Table 11 shows the configurations for these experiments. Experiment 1 and 2 do not include the post processing as opposed to Experiment 3 and 4.

Experiment Nr.	Scenario	Algorithm	Mutation Probability	Set-up Optimization	Worker Constraint	Semi-automatic Machines	Steepest Descent
1	1	GA-NS	0.035				
2	2	GA-NS	0.035	✓			
3	1	GA-NS	0.035				
4	2	GA-NS	0.035	✓			

Table 11: Experiment Configurations Set 3

Hypotheses

Before executing the experiments listed in Table 9, Table 10 and Table 11, we formulate our hypotheses on the results of the experiments. Our hypotheses concern the behaviour of the lateness and AST that we expect to occur when we compare the various scenarios. Our hypotheses are as follows.

Hypothesis 1:

"Including set-up optimization results in lower lateness values than not including set-up optimization."

Scenario 2 includes set-up optimization as opposed to Scenario 1. This set-up optimization aims to reduce the lead time by reducing the required set-up time. Due to this lead time reduction, less machine capacity is utilized, so we expect the lateness to reduce.

Hypothesis 2:

"Including set-up optimization results in a lower AST than not including set-up optimization."

Scenario 1 does not consider set-up times. Scenario 2 however aims to reduce the set-up time. For sure the set-up time cannot increase, because there is nothing that can cause to schedule more set-up time. Regarding set-up times, Scenario 1 is the worst case scenario.

When we compare including set-up optimization to not including set-up optimization we expect to see a decrease in AST, Hypothesis 2. We think however that this decrease, i.e. improvement of the AST, is actually less than this comparison shows. In reality it might occur that, even though set-up optimization is not included, two items with the same set-up parameters happen to follow each other on the same machine. When this occurs by chance the required set-up time decreases. So, when we account for this event to happen for the case where we do not consider set-up optimization, we expect the difference between the two cases, with and without set-up optimization, to be smaller than when we do not account for this event to happen. For this hypothesis we are interested in Set 2 which includes the post processing.

Hypothesis 3:

"By including the worker constraint the created schedule results in a worse lateness than neglecting this constraint."

Scenario 3 includes workers as an additional resource constraint. This is likely to give a worse lateness than both Scenario 1 and 2, because a capacity constraint is added. For this hypothesis we are interested in Set 1, so we do not consider the post processing. We want to know the impact on the schedule.

Hypothesis 4:

"By including the worker constraint in the scheduling process it is possible to achieve better schedules in practice as opposed to not including this constraint, but evaluating the objective as if the workers are a constraint."

If we include the workers as a resource constraint in the scheduling process it is very likely the lateness increases when we compare this to the scheduling process that does not account for the worker constraint. If we however look at the situation at the shop floor at PM, the workers are in fact a constraint for the production process. If we account for this constraint after the scheduling process (Set 2) and then again compare the scheduling scenario that does include the worker constraint during optimization to the scenario that does not include the worker constraint during optimization of the schedule we think the first scenario is capable of achieving better schedules in terms of lateness.

Hypothesis 5:
"Including semi-automatic machines results in a lower lateness than including workers without considering semi-automatic machines."

We expect Scenario 4, which includes semi-automatic machines, to improve the lateness of Scenario 3, because by considering semi-automatic machines, we reduce the impact of the worker constraint. Since in this case machines run partially automatic, workers are not required during the processing time of these machine, only during set-up. We expect this to be true for both the case where Scenario 3 and 4 include set-up optimization and the case where they both do not include set-up optimization.

5.4 Results of the Scheduling Algorithm

When applied in practice, the scheduling algorithm will be applied on a regular basis, since the order set to be scheduled changes over time. Because of this we want to review the scheduling algorithm performance for each experiment based on the average objective achieved over 10 different order sets. We do not pick the best performance, because this might give a distorted view of the performance of the scheduling algorithm, because the order set might vary. We compare the average lateness value achieved over the 10 replication for each scheduling scenario. We run the experiments on a laptop with an intel Core i5 of the 8th generation and 8 GB of RAM memory. We use the program Visual Studio (Community version 2017) where we program the algorithms using Visual Basic. Appendix E: Results, displays the output results of all the experiments for Set 1, Set 2 and Set 3 see respectively Table E1, Table E2 and Table E3.

To answer the hypotheses, we summarize the relevant results in 4 tables. Table 12 Displays the lateness values of the experiments of Set 1. Behind the lateness values we denote the experiment number between brackets. The green bold printed values denote the best performance for each scenario. We abbreviate set-up optimization as "SU".

Algorithm	**Scenario 1:**	**Scenario 2:**	**Scenario 3a:**	**Scenario 3b:**	**Scenario 4a:**	**Scenario 4b:**
	No SU,	**SU,**	**No SU,**	**SU,**	**No SU,**	**SU,**
	No Worker	**No Worker**	**Worker**	**Worker**	**Semi-auto**	**Semi-auto**
GA-VNS	-194.7 (3)	-194.0 (9)	-112.5 (15)	-113.5 (19)	-114.9 (27)	-118.4 (31)
GA-NS	-215.2 (4)	-211.3 (10)	-139.7 (16)	-124.9 (20)	-136.7 (28)	-132.6 (32)
SA	**-232.7** (5)	**-228.5** (11)	**-153.4** (17)	-118.6 (21)	**-157.0** (29)	-129.7 (33)
SD	21.6 (6)	-39.9 (12)	152.2 (18)	147.8 (22)	150.7 (30)	141.9 (34)
GA-VNS + SD	-197.6 (7)	-196.3 (13)	-116.1 (23)	-116.5 (25)	-118.5 (35)	-121.9 (37)
GA-NS + SD	-216.9 (8)	-212.4 (14)	-141.6 (24)	**-128.0** (26)	-138.2 (36)	**-136.3** (38)

Table 12: Lateness Values Set 1

Table 13 shows the lateness values of the experiments of Set 2. Some experiments from Set 1 do not require the post processing. This is the case for the experiments that include both the set-up optimization and the worker constraint. So, applying post processing to these experiments of Set 1 results in the same output values. For completeness we reprint these values for Set 2. Because the experiment numbers conflict for both sets of experiments, we print the copied values from Set 1 in italics. Table 14 and Table 15 respectively present the lateness and AST values of the experiments of Set 3.

Algorithm	**Scenario 1: No SU, No Worker**	**Scenario 2: SU, No Worker**	**Scenario 3a: No SU, Worker**	**Scenario 3b: SU, Worker**	**Scenario 4a: No SU, Semi-auto**	**Scenario 4b: SU, Semi-auto**
GA-VNS	0.7 (1)	48.4 (7)	-112.5 (13)	*-113.5 (19)*	-114.9 (19)	*-118.4 (31)*
GA-NS	-16.6 (2)	30.8 (8)	-139.7 (14)	*-124.9 (20)*	-136.7 (20)	*-132.6 (32)*
SA	-28.2 (3)	22.3 (9)	-153.5 (15)	*-118.6 (21)*	-157.0 (21)	*-129.7 (33)*
SD	181.3 (4)	183.9 (10)	152.2 (16)	*147.8 (22)*	150.7 (22)	*141.9 (34)*
GA-VNS + SD	0.8 (5)	47.2 (11)	-116.1 (17)	*-116.5 (25)*	-118.5 (23)	*-121.9 (37)*
GA-NS + SD	-16.8 (6)	30.6 (12)	-141.6 (18)	*-128.0 (26)*	-138.2 (24)	*-136.3 (38)*

Table 13: Lateness Values Set 2

Algorithm	**Scenario 1: No SU, No Worker**	**Scenario 2: SU, No Worker**
GA-NS (without post processing)	877.8 (1)	568.8 (2)
GA-NS (with post processing)	15070.6 (3)	14582.3 (4)

Table 14: Lateness Values Set 3

Algorithm	**Scenario 1: No SU, No Worker**	**Scenario 2: SU, No Worker**
GA-NS (without post processing)	1.13 (1)	1.10 (2)
GA-NS (with post processing)	1.11 (3)	1.09 (4)

Table 15: AST Values Set 3

Hypotheses Results

Hypothesis 1:

"Including set-up optimization results in lower lateness values than not including set-up optimization."

For this hypothesis we analyze experiments from Set 3, see Table 14. When we take the case where we do not apply post processing we observe a difference in lateness between Experiment 1 and 2 of 309 days. For the case where we do apply the post processing we observe a difference in lateness between Experiment 3 and 4 of 488.3 days. If we evaluate the relative lateness reduction, which we calculate by using 500 as a nominal value, we get a lateness decrease of respectively 61.80% and 97.66%, see Table 16. So when we apply post processing the lateness yields a larger improvement. Additionally we analyze the significance of the difference between the two scenarios, see Table 17. From the p-values of both comparisons we can conclude that the difference for Experiment 1 and 2 is statistically significant for a significance level of 0.05. When we however evaluate the realistic case we see that for Experiments 3 and 4 the difference is not statistically significant, so we cannot conclude for this case that set-up optimization results in a lower lateness.

Algorithm	**Scenario 1 → Scenario 2 Lateness Decrease**
GA-NS (without post processing)	61.80 %
GA-NS (with post processing)	97.66 %

Table 16: Impact Set-up Optimization

H_0	H_1	T-Value	P-Value	Conclusion
$\mu_1 = \mu_2$	$\mu_1 > \mu_2$	2.18	P=0.02	$p < \alpha$, so reject H_0
$\mu_3 = \mu_4$	$\mu_3 > \mu_4$	0.47	P=0.47	$p > \alpha$, so failed to reject H_0

Table 17: Significance Lateness Set-up Optimization

Hypothesis 2:

"Including set-up optimization results in a lower AST than not including set-up optimization."

From Table 15 we see that including set-up optimization (Experiment 2) results in a lower AST than when we do not include set-up optimization (Experiment 1). Also when we apply the post processing we see that there is still a decrease in AST when we apply set-up optimization (Experiment 4) as opposed to not applying set-up optimization (Experiment 3). We also see that there is an AST decrease if we consider the set-up decrease that occurs by chance, compare Experiment 1 and Experiment 3. This means that the AST improvement is in practice less than the schedule suggests. The differences between the ASTs are small, but from Table 18 we see that the differences are statistically significant based on our defined alpha. From the p-values we see that even for a lower significance level the differences would be significantly different. So even though the differences are small, we can say with a confidence level of > 99.95% that these differences are the result of applying set-up optimization and are not caused by coincidence.

Experiments	Difference AST (%)	H_0	H_1	T-Value	P-Value	Conclusion
1 & 2	0.03	$\mu_1 = \mu_2$	$\mu_1 > \mu_2$	31.66	P<0.001	$p < \alpha$, so reject H_0
3 & 4	0.02	$\mu_3 = \mu_4$	$\mu_3 > \mu_4$	12.91	P<0.001	$p < \alpha$, so reject H_0
1 & 3	0.02	$\mu_1 = \mu_3$	$\mu_1 > \mu_3$	16.78	P<0.001	$p < \alpha$, so reject H_0

Table 18: Analysis Set-up Optimization

Hypothesis 3:

"By including the worker constraint the created schedule results in a worse lateness than neglecting this constraint."

If we look at the average lateness values for the experiments of Set 1, see Table 12, we can confirm this hypothesis. We compare the base case, Scenario 1 to Scenario 3a that does include workers but not set-up optimization. Also we compare the case with set-up optimization, Scenario 2, to Scenario 3b that does include both the worker constraint and set-up optimization. In both cases we see that the lateness decreases when we include workers.

For the purpose of comparison it does not matter whether the values are positive or negative. We are only interested in the difference between two lateness values. To compare two lateness values we need a reference value. It does not matter what this value is as long as all values are either larger or smaller than this value. We take the value 500 as a reference value, because we do not observe any lateness value that is smaller than -500 and no value larger than 500. Table 19 displays the resulting differences between Scenario 1 and 3a and Scenario 2 and 3b. From Table 20 we conclude that these differences are all statistically significant for our significance level of 0.05.

Algorithm	**Scenario 1 → Scenario 3a** **Lateness Increase**	**Scenario 2 → Scenario 3b** **Lateness Increase**
GA-VNS	16.4 %	16.1 %
GA-NS	15.1 %	17.3 %
SA	15.9 %	22.0 %
SD	26.1 %	37.5 %
GA-VNS + SD	16.3 %	16.0 %
GA-NS + SD	15.1 %	16.9 %
Average	**17.5 %**	**21.0 %**

Table 19: Impact Worker Constraint

H_0	H_1	T-Value	P-Value	Conclusion
$\mu_3= \mu_{15}$	$\mu_3 < \mu_{15}$	4.66	$P<0.001$	$p < \alpha$, so reject H_0
$\mu_4= \mu_{16}$	$\mu_4 < \mu_{16}$	4.17	$P<0.001$	$p < \alpha$, so reject H_0
$\mu_5= \mu_{17}$	$\mu_5 < \mu_{17}$	3.43	$P=0.002$	$p < \alpha$, so reject H_0
$\mu_6= \mu_{18}$	$\mu_6 < \mu_{18}$	2.41	$P=0.013$	$p < \alpha$, so reject H_0
$\mu_7= \mu_{23}$	$\mu_7 < \mu_{23}$	4.34	$P<0.001$	$p < \alpha$, so reject H_0
$\mu_8= \mu_{24}$	$\mu_8 < \mu_{24}$	4.42	$P<0.001$	$p < \alpha$, so reject H_0
$\mu_9= \mu_{19}$	$\mu_9 < \mu_{19}$	3.79	$P<0.001$	$p < \alpha$, so reject H_0
$\mu_{10}= \mu_{20}$	$\mu_{10} < \mu_{20}$	3.82	$P<0.001$	$p < \alpha$, so reject H_0
$\mu_{11}= \mu_{21}$	$\mu_{11} < \mu_{21}$	4.23	$P<0.001$	$p < \alpha$, so reject H_0
$\mu_{12}= \mu_{22}$	$\mu_{12} < \mu_{22}$	4.16	$P<0.001$	$p < \alpha$, so reject H_0
$\mu_{13}= \mu_{25}$	$\mu_{13} < \mu_{25}$	3.75	$P<0.001$	$p < \alpha$, so reject H_0
$\mu_{14}= \mu_{26}$	$\mu_{14} < \mu_{26}$	3.74	$P<0.001$	$p < \alpha$, so reject H_0

Table 20: Significance Worker Constraint Set 1

Hypothesis 4:

"By including the worker constraint in the scheduling process it is possible to achieve better schedules in practice as opposed to not including this constraint, but evaluating the objective as if the workers are a constraint."

By analysing the lateness values from Set 2 see Table 13, which applies post processing to all experiments, we get the lateness increase in percentage, see Table 21. We again calculate this percentage related to the nominal value 500. The lateness increase is negative which means that the value for the lateness decreases which is an improvement in our objective. For Set 1, recall Hypothesis 3, we see that the lateness value increases on average by 17.5% when we do not include set-up optimization and by 21% when we do include set-up optimization. We cannot conclude anything from these values, because the values are calculated based on an arbitrary nominal value to put the values into perspective. We can however compare the values to the values we present in Table 21, because we use the same nominal value. Set 1 suggests the performance of the schedule decreases when we include the worker constraint. If we look however at the post processed experiments we see that including this constraint actually improves the performance of the created schedules, because the lateness decreases. This means that if the worker constraint is included in the optimization process the algorithm is able to achieve a better schedule in terms of lateness than when this constraint is not included during optimization, but is only applied afterwards. Table 22 shows that for all comparisons, except for Experiment 10 (Set 2) compared to Experiment 22 (Set1), the decrease in lateness is statistically significant for a significance level of 0.05.

Algorithm	Scenario 1 → Scenario 3a Lateness Increase	Scenario 2 → Scenario 3b Lateness Increase
GA-VNS	-22.6 %	-32.4 %
GA-NS	-24.6 %	-31.1 %
SA	-25.1 %	-28.2 %
SD	-5.8 %	-7.2 %
GA-VNS + SD	-23.4 %	-32.7 %
GA-NS + SD	-25.0 %	-31.7 %
Average	**-21.1 %**	**-27.2 %**

Table 21: Impact Post Processing Worker Constraint

H_0	H_1	T-Value	P-Value (p)	Conclusion
$\mu_1 = \mu_{13}$	$\mu_1 > \mu_{13}$	4.88	P<0.001	$p < \alpha$, so reject H_0
$\mu_2 = \mu_{14}$	$\mu_2 > \mu_{14}$	3.73	P<0.001	$p < \alpha$, so reject H_0
$\mu_3 = \mu_{15}$	$\mu_3 > \mu_{15}$	4.87	P<0.001	$p < \alpha$, so reject H_0
$\mu_4 = \mu_{16}$	$\mu_4 > \mu_{16}$	0.42	P<0.001	$p < \alpha$, so reject H_0
$\mu_5 = \mu_{17}$	$\mu_5 > \mu_{17}$	5.13	P<0.001	$p < \alpha$, so reject H_0
$\mu_6 = \mu_{18}$	$\mu_6 > \mu_{18}$	3.78	P<0.001	$p < \alpha$, so reject H_0
$\mu_7 = \mu_{19}$	$\mu_7 > \mu_{19}$	4.91	P<0.001	$p < \alpha$, so reject H_0
$\mu_8 = \mu_{20}$	$\mu_8 > \mu_{20}$	4.44	P<0.001	$p < \alpha$, so reject H_0
$\mu_9 = \mu_{21}$	$\mu_9 > \mu_{21}$	5.16	P<0.001	$p < \alpha$, so reject H_0
$\mu_{10} = \mu_{22}$	$\mu_{10} > \mu_{22}$	0.67	P=0.257	$p > \alpha$, so failed to reject H_0
$\mu_{11} = \mu_{25}$	$\mu_{11} > \mu_{25}$	4.92	P<0.001	$p < \alpha$, so reject H_0
$\mu_{12} = \mu_{26}$	$\mu_{12} > \mu_{26}$	4.57	P<0.001	$p < \alpha$, so reject H_0

Table 22: Significance Worker Constraint Set 2

Hypothesis 5:

"Including semi-automatic machines results in a lower lateness than including workers without considering semi-automatic machines."

For the purpose of answering this hypothesis we analyze the results of Set 1. Again we evaluate the situation where we do not include the set-up times Scenario 3a, without semi-automatic machines and Scenario 4a, with semi-automatic machines and the situation where we include set-up times Scenario 3b and Scenario 4b. For the latter case we see from Table 23 that the impact of including semi-automatic machines is larger than for the first case. Table 24 however shows that none of the observed differences are statistically significant for a significance level of 0.05.

Algorithm	**Scenario 3a → Scenario 4a Lateness Increase**	**Scenario 3b → Scenario 4b Lateness Increase**
GA-VNS	-0.5	-1
GA-NS	0.6	-1.5
SA	-0.7	-2.2
SD	-0.3	-1.2
GA-VNS + SD	-0.5	-1.1
GA-NS + SD	0.7	-1.7
Average	**-0.1 %**	**-1.4 %**

Table 23: Impact Semi-automatic Machines

H_0	H_1	**T-Value**	**P-Value (p)**	**Conclusion**
$\mu_{15}= \mu_{27}$	$\mu_{15} > \mu_{27}$	0.14	P=0.44	$p > \alpha$, so failed to reject H_0
$\mu_{16}= \mu_{28}$	$\mu_{16} < \mu_{28}$	0.17	P=0.44	$p > \alpha$, so failed to reject H_0
$\mu_{17}= \mu_{29}$	$\mu_{17} > \mu_{29}$	0.15	P=0.44	$p > \alpha$, so failed to reject H_0
$\mu_{18}= \mu_{30}$	$\mu_{18} > \mu_{30}$	0.02	P=0.49	$p > \alpha$, so failed to reject H_0
$\mu_{23}= \mu_{35}$	$\mu_{23} > \mu_{35}$	0.15	P=0.44	$p > \alpha$, so failed to reject H_0
$\mu_{24}= \mu_{36}$	$\mu_{24} < \mu_{36}$	0.18	P=0.43	$p > \alpha$, so failed to reject H_0
$\mu_{19}= \mu_{31}$	$\mu_{19} > \mu_{31}$	0.22	P=0.41	$p > \alpha$, so failed to reject H_0
$\mu_{20}= \mu_{32}$	$\mu_{20} > \mu_{32}$	0.34	P=0.37	$p > \alpha$, so failed to reject H_0
$\mu_{21}= \mu_{33}$	$\mu_{21} > \mu_{33}$	0.39	P=0.35	$p > \alpha$, so failed to reject H_0
$\mu_{22}= \mu_{34}$	$\mu_{22} > \mu_{34}$	0.11	P=0.46	$p > \alpha$, so failed to reject H_0
$\mu_{25}= \mu_{37}$	$\mu_{25} > \mu_{37}$	0.25	P=0.40	$p > \alpha$, so failed to reject H_0
$\mu_{26}= \mu_{38}$	$\mu_{26} > \mu_{38}$	0.37	P=0.36	$p > \alpha$, so failed to reject H_0

Table 24: Significance Semi-automatic Machines Set 1

For this hypothesis we only evaluate Set 1, because all experiments under review here do include the worker constraint. We observe from Table 12 and Table 13 that the post processing has most impact for the experiments that do not include the worker constraint. Since the values of Set 1 and Set 2 for the experiments we analyze here are almost completely identical we do not additionally evaluate the effect of the post processing. This effect will be very close to zero.

Key Performance Indicators

For the main results we focus on the lateness, which is also the objective in the scheduling algorithms. Also we consider the AST. We additionally are interested in the performance of the other KPIs. For this analysis we use the second set of results, because these experiments represent reality better than Set 1. It is important to note that the absolute values do not tell us anything, because these values depend highly on the input we provide for the algorithms. For example the due dates and available machine and worker capacity. If we set a higher value for the available machine capacity this will result in higher performance amongst others in terms of lateness and OTDP. We can however use the results for comparison between scenarios and scheduling algorithms. We again apply 500 as the nominal value. For the OTDP we desire a high value, so an increase of this value is an improvement. For the ADL the lower the value the better, so a decrease of this value is an improvement. We also want the AQT to be as low as possible, so a decrease of the AQT value is an improvement.

Table 25 shows the results for comparing the base case (Scenario 1) to including set-up time (Scenario 2). From these results we can conclude that we achieve the desired results for each of the KPIs besides AST.

Algorithm	**OTDP Increase**	**ADL Decrease**	**AQT Decrease**
GA-NS (with post processing)	0.1 %	0.8 %	1.0 %

Table 25: Impact KPIs Set-up Optimization

For including the worker constraint (Scenario 3a) as opposed to the base case (Scenario 1), we observe an improvement for each of the KPIs, see Table 26.

Algorithm	**OTDP Increase**	**ADL Decrease**	**AQT Decrease**
GA-VNS	2.3 %	3.3 %	1.5 %
GA-NS	4.9 %	6.4 %	1.6 %
SA	4.1 %	6.5 %	1.7 %
SD	0 %	0.5 %	0.4 %
GA-VNS + SD	2.5 %	3.5 %	1.6 %
GA-NS + SD	4.9 %	6.5 %	1.7 %
Average	**3.1 %**	**4.5 %**	**1.4 %**

Table 26: Impact KPIs Worker Constraint

When we include the semi-automatic machine extension in the scheduling algorithm, Table 27 shows the impact on each of the KPIs. We observe that for the inclusion of semi-automatic machines the impact on the KPIs is smaller than when we include the worker constraint. We also observe that the GA-NS and the GA-NS+SD results in a decreased performance of the AQT.

Algorithm	**OTDP Increase**	**ADL Decrease**	**AQT Decrease**
GA-VNS	0.4 %	0.1 %	0.04 %
GA-NS	0.4 %	0.2 %	-0.04 %
SA	0.3 %	0.02 %	0.04 %
SD	0 %	0.04 %	0.02 %
GA-VNS + SD	0.4 %	0.1 %	0.02 %
GA-NS + SD	0.5 %	0.2 %	-0.04 %
Average	**0.3 %**	**0.1 %**	**0.01 %**

Table 27: Impact KPIs Semi-automatic Machines

5.5 Conclusion

In answering Sub Question 3b: **“***What should a new and optimal scheduling process for PM look like?”* and Sub Question 4, *“What is the impact of the scheduling improvements for PM?”,* we review several scheduling algorithms, the GA-VNS, GA-NS, SA, SD, GA-VNS+SD and GA-NS+SD. We consider these algorithms in several scenarios. We review the base case (Scenario 1), set-up optimization (Scenario 2), worker constraint (Scenario 3) and semi-automatic machines (Scenario 4).

From the results of the experiments, we see that the impact of set-up optimization is limited. Including this scheduling extension does reduce the lateness, but for the experiments that include post processing the chance is quite high that this effect is not strictly the result of set-up optimization. Including set-up optimization does however reduce the AST. This does not directly reduce the OTDP, but is does reduce the lead time. Reduced lead times make it possible to deliver faster which is an important aspect of customer service next to OTDP. Including the worker constraint reduces the lateness both when set-up optimization is included and when this is not included. We also see that each scheduling algorithm is capable of improving the schedule quality in terms of lateness when this constraint is considered in the scheduling algorithm. Considering semi-automatic machines slightly reduces the lateness. These results are however not statistically significant.

By evaluating the performance of each scheduling algorithm on each of the scenarios, we observe that both SA and GA-NS+SD perform best. So, for PM the extensive GA-VNS algorithm does not perform best as we expected. In case we include all scheduling extensions, set-up optimization, worker constraint and semi-automatic machines, the GA-NS+SD performs best. We clearly see that the SD, which is the most straightforward scheduling algorithm performs worst for each scenario. If we consider the SD as additional procedure after applying the GA-VNS or GA-NS we see that this addition slightly improves the performance of these algorithms.

Including workers is the most important adaption to the scheduling algorithm. Not just because this represents the situation at the shop floor more realistically, but also because including this constraint in the optimization process the scheduling algorithm is capable of achieving a better performance. Including set-up optimization is effective in reducing the set-up times.

6 Conclusions and Recommendations

To achieve the goal of this research to gain knowledge on what the opportunities are in optimizing the scheduling process to create more realistic and improved schedules which will increase the OTDP at PM we answer our research questions. Recall the main question of this research:

"How can PM improve its production schedules to increase the on-time delivery performance?"

We provide the answer to this research question by discussing the sub questions one by one in Section 6.1. Section 6.2 describes our recommendations for PM. Section 6.3 discusses our contribution to the current literature. Section 6.4 provides our thoughts on interesting further research. Section 6.5 presents our critical assessment of this research where we reflect on the choices we make and do improvement suggestions in case we would execute this research again.

6.1 Main Findings

In answering the first sub question "*What is the performance of the currently generated schedules?*" we analyze the current situation at PM. PM produces products both based on an ETO, MTO and MTS strategy. This results in a high variety of products since some products are standard, while others are custom made. Also typical for such a production environment, considering the ETO and MTO production strategies, are the low production volumes. Considering these characteristics we can conclude a job-shop is the best suited production lay-out for PM. This is also the current lay-out for the production departments at PM.

To create the production schedules, PM uses a scheduling tool called Factory Planning. This tool communicates with the ERP system, Glovia to create the schedules. Factory Planning is partially a black box, which makes it hard to evaluate its performance. Although we cannot judge the quality of the core scheduling algorithm of Factory Planning we observe that Factory Planning is an extensive tool with a lot of options and extensions that are relevant for PM. Some interesting Factory Planning extensions are the use of scheduling groups to make the scheduling process more detailed and including labor and tools as capacity resources. This will help to make the schedule more realistic. If schedules are realistic, this will in turn make the schedules easier to realize and contribute to an increased OTDP.

Four KPIs measure the performance of the currently generated schedules. The OTDP, ADL, AQT and the AST are the KPIs that are relevant to assess how realizable and efficient the production schedules created at PM are. Being realizable contributes to be able to deliver the order on the date agreed with the customer. Being efficient contributes to shorter lead times which is important to be able to deliver fast which is also an important contributor to the service for the customer besides delivering on time. PM currently has an OTDP of 80.3%. The trend is that this value is decreasing. The ADL of the orders shipped in 2018 is 17.6 days. This value is increasing over time. The AST of 2018 is 8.4% of the total processing time. The AST is decreasing. The current AQT at the production departments of PM is 7.6 days, and increasing over time. Three KPIs show a trend towards the undesirable direction. Only the AST is improving.

The answer to Sub Question 2 **"***What methods for achieving an optimized production schedule in an MTO environment are described in literature?"* is as follows. Literature proposes various scheduling algorithms for the DRCFJSP. Literature often proposes the GA for the JSP and FJSP. Literature on the DRCFJSP is more scarce, but we also find promising results for the GA applied to the DRCFJSP. Recent literature proposes the GA-VNS. Literature proposes to encounter earliness to create robust schedules against any disturbances that may occur at the shop floor. Unforeseen disturbances can cause an order to be delivered too late. By using minimizing lateness as the objective function earliness is encountered as negative lateness. This slack time works as a buffer to compensate for time loss due to for example machine break down.

Sub Question 3 "*What are potential scheduling improvements for PM?"* consists of two parts. Sub Question 3a *"How can existing extensions of the planning system of PM be implemented?"* considers Factory Planning. Set-up optimization is an interesting scheduling extension that can be gradually implemented to Factory Planning. Besides the set-up time reduction that this extension can achieve, it is important to include this extension because Factory Planning does not detect coincidental set-up optimization which reduces the realistic representation of the situation on the shop floor.

In addressing Sub Question 4 "*What is the impact of the scheduling improvements for PM?"* for Factory Planning we find that set-up optimization can yield a set-up time reduction of 9.5%. We however expect this effect to be larger if more items and more machines are included in the set-up optimization.

For Sub Question 3b: **"***What should a new and optimal scheduling process for PM look like?"*, we consider 6 alternative scheduling algorithms: the GA-VNS, GA-NS, SA, SD, GA-VNS+SD and GA-NS+SD. For most scheduling scenarios the SA performs best, except for Scenario 3b and 4b the GA-NS+SD performs best.

For Sub Question 4 "*What is the impact of the scheduling improvements for PM?",* related to the alternative scheduling algorithms we see that including set-up optimization is very likely to achieve a reduced AST. The effect of actively optimizing the set-up is however limited because it sometimes occurs that set-up time can be neglected because by chance two sequential items have the same set-up parameter values. Including the worker constraint has most impact. Additionally including semi-automatic machines does not improve the KPIs a lot. Including the worker constraint is most important to include to make the schedules better in terms of lateness and being more realistic. Including the worker constraint and semi-automatic machines both improve the performance on all KPIs. The impact of semi-automatic machines is however small compared to the impact of including the worker constraint.

6.2 Recommendations for Practice

PM currently uses Factory Planning to create the production schedules. This is an extensive scheduling tool that PM uses since 2016. There are a lot of extensions available in Factory Planning that PM has not yet implemented. Not all extensions are however relevant for PM, since Factory Planning is a general tool for a broad range of industries. Next to investigating some of the extensions we also analyze 6 alternative scheduling algorithms. Because we are not able to compare the performance of these algorithms to Factory Planning it is hard to decide for PM whether they would want to invest in setting up a new scheduling algorithm. Because their current scheduling tool is compatible with their ERP system Glovia and Factory Planning offers a lot of possibilities to enhance the quality of the schedules we think for now it is undesirable to invest time and money in a new scheduling algorithm. An additional important argument is that we were not able to compare the alternative scheduling algorithms to Factory Planning, so we cannot say whether any of the alternative algorithms performs better than Factory Planning. Factory Planning is however a non-customized scheduling tool, so it is also interesting for PM to consider investing in a tool that applies SA or GA-NS + SD which can be matched to the exact needs of PM, like including set-up optimization, the worker constraint and semi-automatic machines. For now, PM can achieve an improvement in the OTDP by implementing the Factory Planning extensions, but on the long term it might pay off to invest in a scheduling algorithm that searches for schedules in a possibly smarter way than Factory Planning.

To improve the production schedules in order to increase the on-time delivery performance, we recommend PM on the short term to:

- Implement set-up optimization in their current scheduling process. This will reduce the lead times and make the schedules more realistic.
- Implement the worker constraint in Factory Planning. From the experiments with the alternative algorithms we conclude that this hugely impacts the possibility of realizing the created schedules.
- Consider including semi-automatic machines in the scheduling process. This extension has less impact than including the worker constraint, but including the worker constraint is a prerequisite for including semi-automatic machines to be useful. We show an improvement in the lateness when we include this extension in our experiments. Our experiments are however not strong enough to prove that this improvement is caused by the inclusion of semi-automatic machines, so additional experiments might be desirable.
- Critically reassess the current data in Glovia. Reliable processing and set-up times are a prerequisite for any scheduling improvement to be beneficial.

On the long term, we recommend PM to:

- Conduct additional research on the performance of SA and GA-NS + SD compared to Factory Planning. These algorithms perform best, so it is interesting, if it is possible to get more insight in Factory Planning, to compare these to the scheduling algorithm of Factory Planning. Since we did not succeed unraveling the scheduling algorithm of Factory Planning, it might be very time consuming and maybe even impossible to get more insight in the scheduling algorithm of Factory Planning.
- Conduct additional research on the possibilities of connecting a new scheduling algorithm to Glovia. To be able to create a schedule by a new scheduling algorithm, information on, amongst others, processing times are required. This information is available from the ERP, Glovia, so it is important for PM that any new scheduling algorithm is compatible to Glovia.

Depending on the results of this additional research, PM can, after improving the performance of Factory Planning, improve the performance of the schedules even more by considering an alternative scheduling algorithm that also includes the scheduling extensions that we recommend to implement for Factory Planning and also applies a smarter scheduling logic that is custom-made for PM.

6.3 Contribution to Literature

This research contributes to the current knowledge on DRCFJSP. We provide insight in the importance of creating schedules that represent reality. We also show the effect of including a secondary constraint on the lateness of a schedule and on the extent to which it is possible to realize a schedule. Current literature mostly applies the makespan as the objective function. We critically assess our choice for what objective to use and provide insight in the results when lateness is the objective in the DRCFJSP.

In this research we combine several real-life aspects in a manufacturing setting such as at PM to improve the scheduling process. By doing so we contribute to the current knowledge on scheduling problems by combining the DRCFJSP and semi-automatic machines. This adds a dimension to scheduling problems next to the DRC.

This research also has value for science in the field of scheduling algorithms. We compare among 6 different scheduling algorithms and we show how they all perform in the case of PM. Because we built the algorithms based on the intermediate results, we were able to determine what aspects of each algorithm performs well in our case. We started off with the GA-VNS based on our conclusion from our literature review. For comparative purposes we also include SA. From the preliminary experiments for each of these algorithms we decide to implement the local search procedure of SA in the GA, resulting in the GA-NS. This new algorithm performs better than the GA-VNS. By additionally applying SD to the GA-VNS and the GA-NS the algorithms perform even better. Because the algorithms partially overlap, the experiments we execute provide interesting insights on what part of the algorithm results in an improved performance. For example by comparing the GA-VNS to the GA-VNS + SD we can conclude that the improvement of this second algorithm is due to the additional SD procedure.

6.4 Further Research

Based on the knowledge we gained by conducting this research it is interesting both for PM and current literature to extend the evaluated scheduling algorithms with enhanced set-up optimization procedures. Also multiple workforce details are interesting to take into account in further research, for example the learning effect, varying effectiveness among workers and illness.

In this research we focus on minimizing the lateness. This decision is based on the idea of including buffer time into the schedule which reduces the impact of unforeseen events such as machine breakdown or delay during processing. Minimizing lateness might result in earliness. This is desirable to account for the unforeseen events. Too much earliness might however cause additional holding costs, because orders are finished too early. This trade-off between robustness against unforeseen events and holding costs is interesting to account for in future research.

6.5 Discussion

In setting up this research we aimed to compare our proposed algorithms to the scheduling algorithm of Factory Planning. We spent a lot of time on finding out the logic that Factory Planning currently uses. Unfortunately we did not achieve this, but a lot of decisions are influenced by this initial intention. If we knew on beforehand we could not make the comparison between the proposed algorithms and Factory Planning we would have been able to make better decisions in this research and spend more time on setting up a stronger experiment. A few of these decisions are the creation of the order sets and determining the due dates for our orders. We focussed too much on creating a set of orders that also is present in Factory Planning.

We assumed that minimizing the lateness would improve the OTDP. From our results of the alternative algorithms we find that the OTDP indeed improves along with the lateness, but does not improve as much as we expected. This can be due to the experiment inputs, for example defining a later due date will improve the performance of the schedule, but it could also be due to the fact that lateness influences the OTDP less than expected. In this case it would have been interesting to include the OTDP as a constraint or use this as the objective function.

From the KPI analysis on the current situation we see that the AST is decreasing over time. This is positive, because set-up time is required, but does not add value to the end product. So it is desirable to minimize the set-up time. It is however unclear what causes this decrease. There has been a change however at the grinding department. The set-up times were defined in the ERP separate from the processing times. Because it turned out to be too time consuming for the workers of the grinding department to separately register the set-up times and the processing times, PM decided to include the set-up time in the processing time. This of course reduces the AST. This could be a cause of the downward trend we observe for the AST, recall Figure 14. This adjustment in the ERP is however done at one moment, so the AST reduction should cause a peak. Besides the peaks in the graph in Figure 14 we still see a steady downward trend. The fact that the foremen of the production departments manually combine operations to reduce the set-up time, could be another explanation for this downward trend. Although it is more likely that this set-up time reduction would be on average the same. It could be however some sort of learning effect, that the foremen get more innovative in combining operations and communicate this to the planner, who can manually change the sequence of operations in the schedule. This could cause the downward trend. Even though the AST already decreases, there is a need to include set-up optimization in the scheduling process, because doing it manually like the foremen currently do, does decrease the AST, but it would be more convenient if the scheduling algorithm does it automatically.

It is quite remarkable that the GA-VNS does not turn out to be the best algorithm for PM. This is the most extensive algorithm and we also applied parameter tuning only for this algorithm. Still this algorithm did not manage to outperform all the other algorithms. The GA-VNS only outperforms SD. We see that adding SD to both the GA-VNS and GA-NS is capable of slightly improving the GA-VNS and GA-NS without SD. It makes sense that the GA-VNS does not outperform the GA-VNS extended with SD, because the additional SD procedure will for sure not make the performance worse. An explanation for the fact that SA performs very well could be that the number of local search iterations is larger for SA than for GA-VNS. Although the GA-NS also outperforms the GA-VNS, so the number of iterations is not the only explanation for the improved performance of the SA compared to GA-VNS. On the other hand we also cannot conclude that the local search procedure is the only part of the SA that improves the algorithm, because SA also outperforms the GA-NS that consists of the same local search procedure.

7 References

Akbar, M., & Irohara, T. (2018). Dual Resource Constrained Scheduling Considering Operator Working Modes and Moving in Identical Parallel Machines Using a Permutation-Based Genetic Algorithm. *International Federation for Information Processing*, 464-472. doi:10.1007/978-3-319-99704-9_57

Al-Hinai, N., & ElMekkawy, T. Y. (2011). Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *International Journal of Production Economics, 132*(2), 279-291. doi:10.1016/j.ijpe.2011.04.020

Amjad, M. K., Butt, S. I., Kousar, R., Ahmad, R., Agha, M. H., Faping, Z., . . . Asgher, U. (2018). Recent Research Trends in Genetic Algorithm Based Flexible Job Shop Scheduling Problems. *Mathematical Problems in Engineering*, 1-32. doi:10.1155/2018/9270802

Bobrowski, P. M., & Park, P. S. (1993). An evaluation of labor assignment rules when workers are not perfectly interchangeable. *Journal of Operations Management, 11*(3), 257-268. doi:10.1016/0272-6963(93)90003-8

Çaliş, B., & Bulkan, S. J. (2015). A research survey: review of AI solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing, 26*(5), 961-973. doi:10.1007/s10845-013-0837-8

Calzarossa, M. C., Vedova, M. L., Massari, L., Nebbione, G., & Tessera, D. (2019). Tuning Genetic Algorithms for resource provisioning and scheduling in uncertain cloud environments: Challenges and findings. *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 174-180. doi:10.1109/PDP.2019.00031

Dhiflaoui, M., Nouri, H. E., & Driss, O. B. (2018). Dual-Resource Constraints in Classical and Flexible Job Shop Problems: A State-of-the-Art Review. *Procedia Computer Science*(126), 1507-1515. doi:10.1016/j.procs.2018.08.123

Di Gaspero, L. (2003). Local Search Techniques for Scheduling Problems: Algorithms and Software Tools. *PHD Thesis*.

Dowsland, K. A., & Thompson, J. M. (2012). Simulated Annealing. *Handbook of Natural Computing*, 1623-1655. doi:10.1007/978-3-540-92910-9_49

ElMaraghy, H., Patel, V., & Abdallah, I. B. (2000). Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms. *Journal of Manufacturing Systems, 19*(3), 186-201. doi:10.1016/S0278-6125(00)80011-4

Felan, J. T., & Fry, T. D. (2001). Multi-level heterogeneous worker flexibility in a Dual Resource Constrained (DRC) job-shop. *International Journal of Production Research, 39*(14), 3041-3059. doi:10.1080/00207540110047702

Fortnow, L. (2009). The Status of the P Versus NP Problem. *Communications of the ACM, 52*(9), 78-86. doi:10.1145/1562164.1562186

Gademan, A. J., & Schutten, J. M. (2002). Linear programming based heuristics for multi-project capacity planning. *BETA publicatie: working papers*(79).

Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research, 1*(2), 117-129. Retrieved from https://www.jstor.org/stable/3689278

Hans, E. W., Herroelen, W., Leus, R., & Wullink, G. (2007). A hierarchical approach to multi-project planning under uncertainty. *Omega, 35*(5), 563-577. doi:10.1016/j.omega.2005.10.004

Hartmann, S., & Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research, 127*(2), 394-407. doi:10.1016/S0377-2217(99)00485Jensen, M. T. (2001). Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures. *Applied Soft Computing, 1*(1), 35-52. doi:10.1016/S1568-4946(01)00005-9

Kher, H. V., & Fredendall, L. D. (2004). Comparing variance reduction to managing system variance in a job shop. *Computer & Industrial Engineering, 46*(1), 101-120. doi:10.1016/j.cie.2003.11.002

Kher, H. V., & Fry, T. D. (2001). Labour felxibility and assignment policies in a job shop having incommensurable objectives. *International Journal of Production Research, 39*(11), 2295-2311. doi:10.1080/00207540110036704

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research, 174*, 23-37. doi:10.1016/j.ejor.2005.01.065

Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety, 91*(9), 992-1007. doi:10.1016/j.ress.2005.11.018

Krajewski, L. J., Malhotra, M. K., & Ritzman, L. P. (2016). *Operations Management* (11 ed.). Essex: Pearson Education Limited.

Land, M., & Gaalman, G. (1996). Workload control concepts in job shops A critical assessment. *International Journal of Production Economics*(46), 535-548. doi:10.1016/S0925-5273(96)00088-6

Larsen, R. L., & Marx, M. L. (2012). *An Introduction to Mathematical Statistics* (5th ed.). Boston: Pearson .

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H., & Shmoys, D. B. (1993). Sequencing and Scheduling: Algorithms and Complexity. *Handbooks in Operaions Research and Management Science, 4*, 445-522. doi:10.1016/S0927-0507(05)80189-6

Liu, L., Gu, H., & Xi, Y. (2007). Robust and stable scheduling of a single machine with random machine breakdowns. *The International Journal of Advanced Manufacturing Technology, 31*(7-8), 645-654. doi:10.1007/s00170-005-0237-0

Morinaga, Y., Nagao, M., & Sano, M. (2014). Optimization of Flexible Job-shop Scheduling with Weighted Tardiness and Setup-Worker Load Balance in Make-to-order Manufacturing. *Joint 7th International Conference on Soft Computing and Intelligent Systems and 15th International Symposium on Advanced intelligent Systems*. doi:10.1109/SCIS-ISIS.2014.7044681

Naderi, B., Zandieh, M., & Fatemi Ghomi, S. M. (2009). Scheduling job shop problems with sequence-dependent setup times. *International Journal of Production Research, 47*(21), 5959-5976. doi:10.1080/00207540802165817

Osman, I. H. (2003). Focused issue on applied meta-heuristics. *Computers & Industrial Engineering, 44*(2), 205-207. doi:10.1016/S0360-8352(02)00175-4

Paprocka, I., Gwiazda, A., & Bączkowicz, M. (2017). Robustness of Schedules Obtained Using the Tabu Search Algorithm Based on the Average SLack Method. *International Joint Conference, 527*, 426-434. doi:10.1007/978-3-319-47364-2_41

Pellerin, R., Perrier, N., & Berthaut, F. (2019). A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operation Research*. doi:10.1016/j.ejor.2019.01.063

PM. (2019a). *Innovation*. Retrieved February 20, 2019, from PM: https://www.pm.nl/about-pm/innovations

PM. (2019b). *About PM*. Retrieved February 26, 2019, from PM: https://www.pm.nl/about-pm

PM. (2019c). *RSDE linear bearing set*. Retrieved from pm: https://www.pm.nl/en/linear-bearing-sets-rsde

Roeva, R., Fidanova, S., & Paprzycki, M. (2013). Influence of the Population Size on the Genetic Algorithm Performance in Case of Cultivation Process Modelling. *Proceedings of the 2013 Federated Conference on Computer Science and Information Systems*, 371-376.

Sanches, D. S., Rocha, J., Castoldi, M. F., Morandin, O., & Kato, E. R. (2015). An Adatptive Genetic Algorithm for Production Scheduling on Manufacturing Systems with Simultaneous Use of Machines and AGVs. *Journal of Control, Automation and Electrical Systems, 26*(3), 225-234. doi:10.1007/s40313-015-0174-6

Silver, E. A. (2004). An overview of heuristic solution methods. *Journal of the Operatinal Research Society, 55*, 936-956. doi:10.1057/palgrave.jors.2601758

Slack, N., Brandon-Jones, A., & Johnston, R. (2013). Operations Management. Harlow, United Kingdom: Pearson.

Teo, C.-C., Bhatnagar, R., & Graves, S. C. (2012). An Application of Master Schedule Smoothing and Planned Lead Time Control. *Production and Operations Management, 21*(2), 211-223. doi:10.1111/j.1937-5956.2011.01263.x

Thürer, M. (2018). Dual Resource Constrained (DRC) Shops: Literature Review and Analysis. *Advances in Production Management Systems*, 81-86. doi:10.1007/978-3-319-99704-9_10

Verbeeck, C. (2011). Heuristische oplossingsmethoden voor het time-constrained project scheduling problem. *Master thesis*, Universiteit Gent.

Wu, R., Li, Y., Gou, S., & 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), 1-14. doi:10.1177/1687814018804096

Xu, J., Xu, X., & Xie, S. Q. (2011). Recent developments in Dual Resource Constrained (DRC) system research. *European Journal of Operational Research, 215*(2), 309-318. doi:10.1016/j.ejor.2011.03.004

Xu, S., & Bean, J. C. (2016). Scheduling parallel-machine batch operations to maximize on-time delivery performance. *Journal of Scheduling, 19*(5), 583-600. doi:10.1007/s10951-015-0449-6

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

Zarandi, M. H., Asl, A. A., Sotudian, S., & Castillo, O. (2018). A state of the art review of intelligent scheduling. *Artificial Intelligence Review*, 1-93. doi:10.1007/s10462-018-9667-6

Ziaee, M. (2014). A heuristic algortihm for solving flexible job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 519-528. doi:10.1007/s00170-013-5510-z

Zijm, W. (2000). Towards intelligent manufacturing planning and control systems. *OR Spektrum, 22*, 313-345. doi:10.1007/s002919900032

Appendix A: Factory Planning Details

Scheduling Direction – Forward: FP starts scheduling at either the current date or at the current date plus the number of days defined at the "Horizon Start Offset" until the current date plus "Horizon End Offset" is reached.

Scheduling Direction – Backward: FP starts at the end of the planning horizon, which is the current date plus "Horizon End Offset" until it reaches the current date plus "Horzion Start Offset"

Horizon Start Offset (from begin of day): the number of days entered here defines when the scheduling process starts. The field below defines the point in time of that day when scheduling is started in minutes. This is currently defined as the starting of the workday which is at 07:30 AM.

Horizon End Offset: defines when FP should stop the scheduling process as the current date plus the number of days entered in this field.

Selection Horizon for Due Date: only orders with a due date that falls within the current date and the current date plus the number of days entered in this field.

Choose longterm/shortterm: determines whether only shortterm orders, longterm orders or both should be scheduled.

Longterm offset: determines when an order is defined as either an longterm or shortterm order. If the ERP End Date of an order is beyond the current date plus the number of days entered in this field, the order is defined as a longterm order, otherwise, the order is defined as a shortterm order.

Start with WC within order: this is used when a certain bottleneck resource (machine) should be scheduled first before all other operations are scheduled around this resource.

Schedule Only Unplanned Operations: defines whether all orders should be (re)scheduled regardless whether an order was already scheduled or not.

Initialize all resources: defines whether all resources should be emptied when the scheduling is started. When all resources are initialized then no order is scheduled on any machine. This field should be unticked if "Schedule Only Unplanned Operations" is selected.

Not plan offset: can freeze a defined period. Operations from the previous scheduling run, that are scheduled during this period are not rescheduled. This option is only applicable if the scheduling direction is forward.

Fix operation sequence offset: if set to "ON", operations can receive a new scheduled date, but the sequence of the orders scheduled within the defined timeframe is maintained.

Machines: this field defines whether the machine capacity is encountered as a constraint. This is done by setting the field to either finite or infinite. For some machines an overload is defined, which means that the regular capacity might be exceeded. The overload can also be set to either infinite or finite.

Labour Resources: this field defines whether the labour capacity is encountered as a constraint. This can only be set if all input on the labour skills and team planning are filled in.

Use labour resource in range: this option is only relevant if all input on the labour skills and team planning are filled in. This is not the case for PM, so we will not elaborate on this setting.

Material: this field defines whether the available material is encountered as a constraint.

Tools: this field defines whether the availability of tools is encountered as a constraint.

Use MRP Start Date: if this field is selected, the first operation of the order will not be scheduled before the MRP start date. This option is only available if the scheduling direction is forward.

Use MRP Due Date: if this field is selected the last operation of the order will not be scheduled such that it will finish before the MRP Due Date. This option is only available if the scheduling direction is backward.

Use Pegging Dates: if this field is selected pegging of the order is supported.

Setup Optimization – no optimization: FP does not apply the setup optimization logic.
Setup Optimization – optimization: the setup logic that aims to minimize the setup times is applied. For this optimization to be of any impact, the required set-up input data needs to be filled in.
Setup Optimization – optimize & suppress SU-time: FP only considers the set-up and run times for the first operation.

Use operation start/end dates: this option is relevant when multiple scheduling idents are run which results in a combination of forward and backward scheduling. The backward scheduling direction determines the earliest start date. The next schedule ident, if rescheduling scheduled orders is allowed, must not start the operation before this date. If this field is selected, operations will not be scheduled before the found earliest start date.

Strategies: in this field, one or multiple scheduling strategies can be hierarchically defined. In FP there are some predefined scheduling strategies available, but the user can also define new scheduling strategies. Prior to executing these strategies in the given order, FP checks whether setup optimization is checked, if so, FP runs the setup optimization as the first strategy.

Schedule Operation with status "In Process" – Schedule and ignore status: regardless of the status the operations will be scheduled. So, also operations with the status of "in process" will be (re)scheduled.
Schedule Operation with status "In Process" – Do not schedule: operations that are "in process" will not be (re)scheduled.
Schedule Operation with status "In Process" – Enforce as first operation: Operations that are "in process" will be scheduled at the beginning of the scheduling horizon, whether the MRP start date is in the future is ignored.

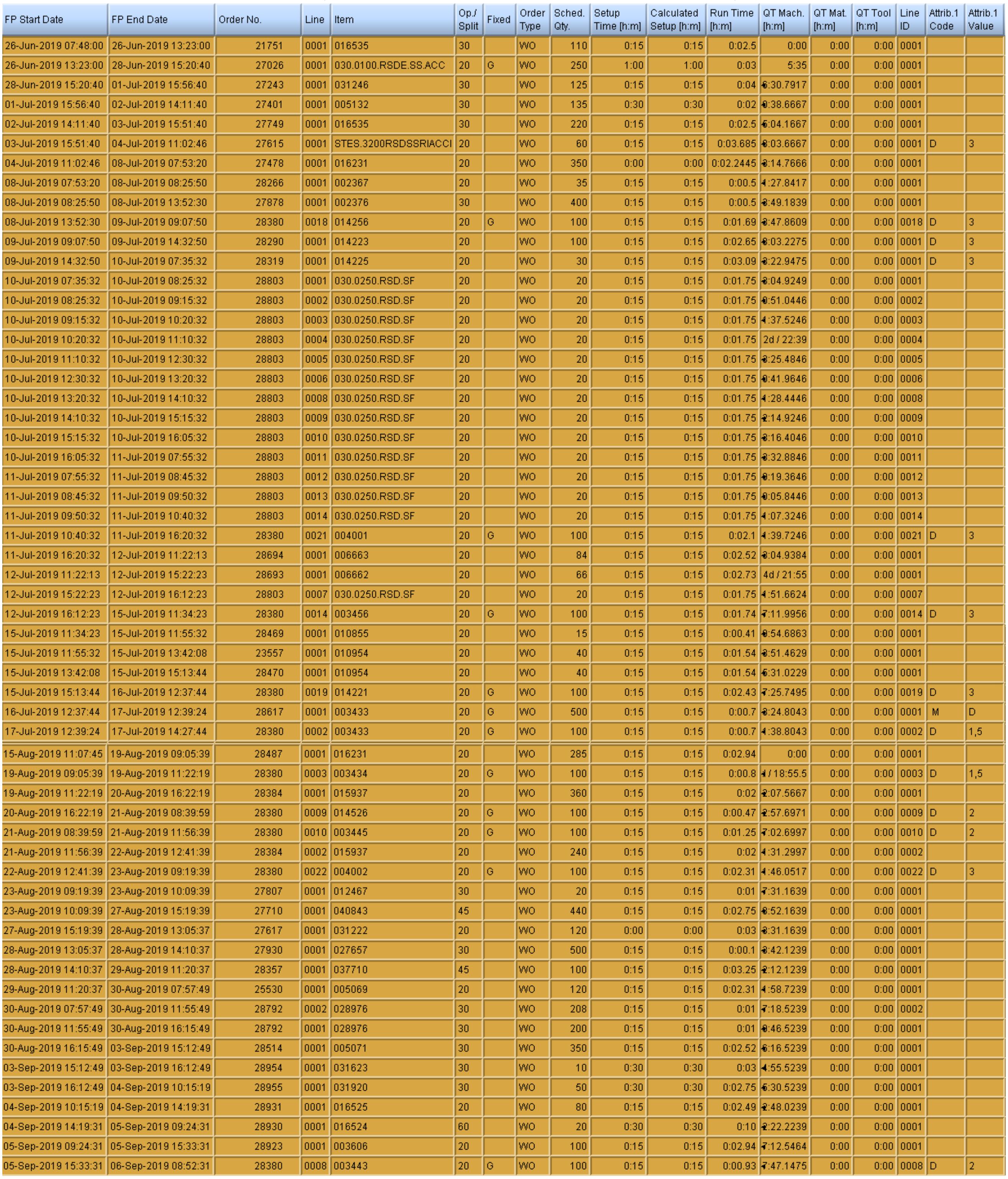
Truncate operations to ERP Dates: this option can only be used if the "Use MRP Start date" or "Use MRP Due Date" field is selected. The purpose of this setting is to try to match the schedule to the MRP dates. Whenever MRP dates are in the past, the operation time is reduced such that MRP dates are met.

Allocate Materials When Unplanned: if this field is checked, FP allocates the material quantity in case the available material is insufficient. This setting only has added value if the material setting is finite.

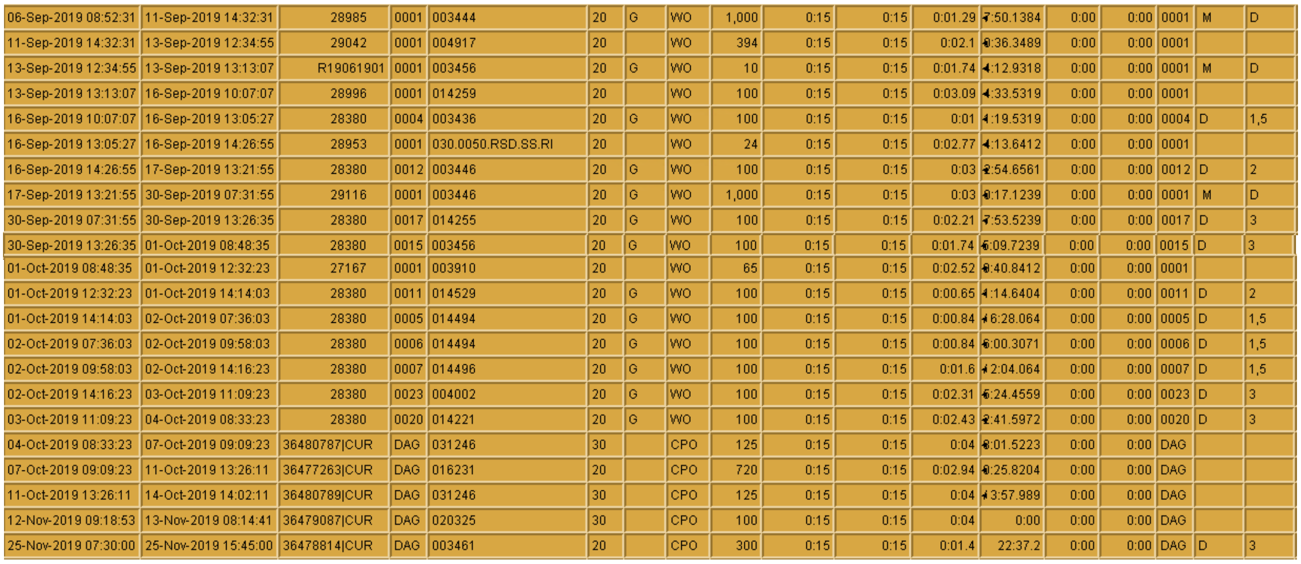
Schedule Ident:	*Veryshortforward*	*Shortforward*	*Longbackward*
Scheduling direction	Forward	Forward	Backward
Horizon Start Offset (time of day)	0 (07:30)	2 (07:30)	0 (07:30)
Horizon End Offset	1000	1000	1000
Selection horizon for Due Date	1000	1000	1000
Longterm/Shortterm (offset in days)	Shortterm (30)	Shortterm (365)	Both (N/A)
Start With WC within order	All work centers	All work centers	All work centers
Schedule only unplanned operations	[ ]	[√]	[√]
Schedule blocks	[ ]	[ ]	N/A
Item schedule groups	All orders	All orders	All orders
Initialize all resources	[√]	[ ]	[ ]
Not plan offset	OFF	OFF	N/A
Fix operation sequence offset	OFF	OFF	N/A
Machines	Finite & overload infinite	Finite & overload infinite	All infinite
Labour resources	Ignore completely	Ignore completely	Ignore completely
Material	Finite	Finite & Buy infinite	All infinite
Tools	Finite	All infinite	All infinite
Use MRP start date	[ ]	[√]	N/A
Use MRP Due Date	N/A	N/A	[ ]
Use Pegging Dates	[ ]	[ ]	[ ]
Setup Optimization	No optimization	No optimization	N/A
Use operation start/end dates	[ ]	[ ]	[ ]
Strategies	1. Priorities	1. Priorities	1. Priorities
	2. Slack Time (just in time)	2. Slack Time (just in time)	2. Slack Time (just in time)
	3. Order ident, Line, Op., Split	3. Order ident, Line, Op., Split	3. Order ident, Line, Op., Split
	4. MRP End Date	4. MRP End Date	4. MRP End Date
	5. MRP Status	5. MRP Status	5. MRP Status
Schedule operations with status "In Process"	Schedule and ignore status	Schedule and ignore status	Schedule and ignore status
Truncate operations to MRP-dates	[ ]	[ ]	[ ]
Allocate materials when unplanned	[ ]	[ ]	[ ]

Appendix B: Set-up Optimization Results

The operations lists displays all operations that are scheduled to be processed by Machine 4. This list also shows the planned start and end dates and the required set-up time. We include two operation lists for Machine 4. Figure B1 shows the operation list for the scenario without set-up optimization. Figure B2 shows the operation list for the scenario with set-up optimization. It is most interesting to look at column 11 “Calculated Setup”. This column contains the calculated set-up times.

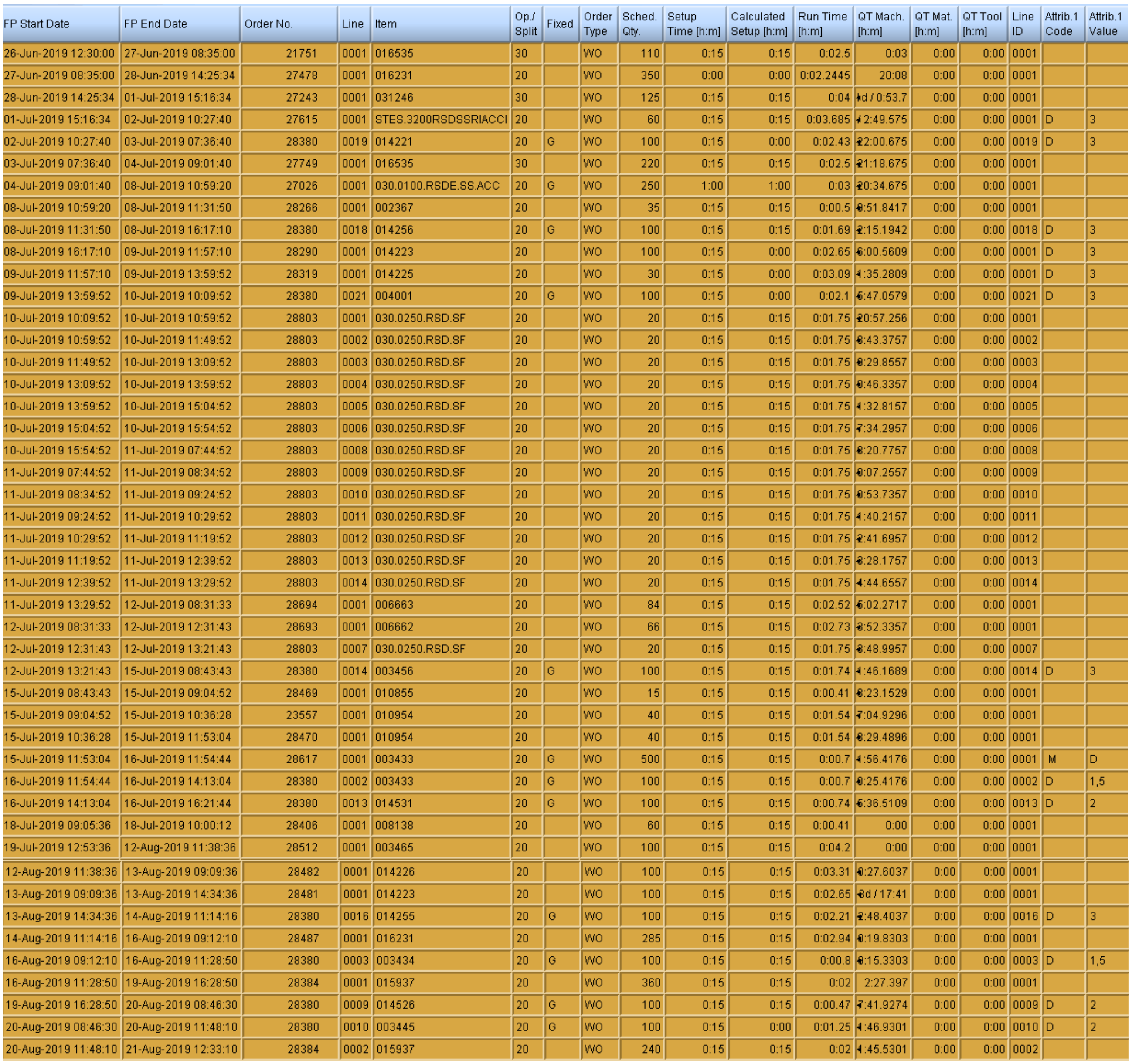


FP Start Date	FP End Date	Order No.	Line	Item	Op./ Split	Fixed	Order Type	Sched. Qty.	Setup Time [h:m]	Calculated Setup [h:m]	Run Time [h:m]	QT Mach. [h:m]	QT Mat. [h:m]	QT Tool [h:m]	Line ID	Attrib.1 Code	Attrib.1 Value
26-Jun-2019 07:48:00	26-Jun-2019 13:23:00	21751	0001	016535	30		WO	110	0:15	0:15	0:02.5	0:00	0:00	0:00	0001		
26-Jun-2019 13:23:00	28-Jun-2019 15:20:40	27026	0001	030.0100.RSDE.SS.ACC	20	G	WO	250	1:00	1:00	0:03	5:35	0:00	0:00	0001		
28-Jun-2019 15:20:40	01-Jul-2019 15:56:40	27243	0001	031246	30		WO	125	0:15	0:15	0:04	6:30.7917	0:00	0:00	0001		
01-Jul-2019 15:56:40	02-Jul-2019 14:11:40	27401	0001	005132	30		WO	135	0:30	0:30	0:02	0:38.6667	0:00	0:00	0001		
02-Jul-2019 14:11:40	03-Jul-2019 15:51:40	27749	0001	016535	30		WO	220	0:15	0:15	0:02.5	6:04.1667	0:00	0:00	0001		
03-Jul-2019 15:51:40	04-Jul-2019 11:02:46	27615	0001	STES.3200RSDSSRIACCI	20		WO	60	0:15	0:15	0:03.685	8:03.6667	0:00	0:00	0001	D	3
04-Jul-2019 11:02:46	08-Jul-2019 07:53:20	27478	0001	016231	20		WO	350	0:00	0:00	0:02.2445	3:14.7666	0:00	0:00	0001		
08-Jul-2019 07:53:20	08-Jul-2019 08:25:50	28266	0001	002367	20		WO	35	0:15	0:15	0:00.5	4:27.8417	0:00	0:00	0001		
08-Jul-2019 08:25:50	08-Jul-2019 13:52:30	27878	0001	002376	30		WO	400	0:15	0:15	0:00.5	3:49.1839	0:00	0:00	0001		
08-Jul-2019 13:52:30	09-Jul-2019 09:07:50	28380	0018	014256	20	G	WO	100	0:15	0:15	0:01.69	3:47.8609	0:00	0:00	0018	D	3
09-Jul-2019 09:07:50	09-Jul-2019 14:32:50	28290	0001	014223	20		WO	100	0:15	0:15	0:02.65	8:03.2275	0:00	0:00	0001	D	3
09-Jul-2019 14:32:50	10-Jul-2019 07:35:32	28319	0001	014225	20		WO	30	0:15	0:15	0:03.09	3:22.9475	0:00	0:00	0001	D	3
10-Jul-2019 07:35:32	10-Jul-2019 08:25:32	28803	0001	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	3:04.9249	0:00	0:00	0001		
10-Jul-2019 08:25:32	10-Jul-2019 09:15:32	28803	0002	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	0:51.0446	0:00	0:00	0002		
10-Jul-2019 09:15:32	10-Jul-2019 10:20:32	28803	0003	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	4:37.5246	0:00	0:00	0003		
10-Jul-2019 10:20:32	10-Jul-2019 11:10:32	28803	0004	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	2d / 22:39	0:00	0:00	0004		
10-Jul-2019 11:10:32	10-Jul-2019 12:30:32	28803	0005	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	3:25.4846	0:00	0:00	0005		
10-Jul-2019 12:30:32	10-Jul-2019 13:20:32	28803	0006	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	0:41.9646	0:00	0:00	0006		
10-Jul-2019 13:20:32	10-Jul-2019 14:10:32	28803	0008	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	4:28.4446	0:00	0:00	0008		
10-Jul-2019 14:10:32	10-Jul-2019 15:15:32	28803	0009	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	2:14.9246	0:00	0:00	0009		
10-Jul-2019 15:15:32	10-Jul-2019 16:05:32	28803	0010	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	3:16.4046	0:00	0:00	0010		
10-Jul-2019 16:05:32	11-Jul-2019 07:55:32	28803	0011	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	3:32.8846	0:00	0:00	0011		
11-Jul-2019 07:55:32	11-Jul-2019 08:45:32	28803	0012	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	8:19.3646	0:00	0:00	0012		
11-Jul-2019 08:45:32	11-Jul-2019 09:50:32	28803	0013	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	0:05.8446	0:00	0:00	0013		
11-Jul-2019 09:50:32	11-Jul-2019 10:40:32	28803	0014	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	4:07.3246	0:00	0:00	0014		
11-Jul-2019 10:40:32	11-Jul-2019 16:20:32	28380	0021	004001	20	G	WO	100	0:15	0:15	0:02.1	4:39.7246	0:00	0:00	0021	D	3
11-Jul-2019 16:20:32	12-Jul-2019 11:22:13	28694	0001	006663	20		WO	84	0:15	0:15	0:02.52	3:04.9384	0:00	0:00	0001		
12-Jul-2019 11:22:13	12-Jul-2019 15:22:23	28693	0001	006662	20		WO	66	0:15	0:15	0:02.73	4d / 21:55	0:00	0:00	0001		
12-Jul-2019 15:22:23	12-Jul-2019 16:12:23	28803	0007	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	4:51.6624	0:00	0:00	0007		
12-Jul-2019 16:12:23	15-Jul-2019 11:34:23	28380	0014	003456	20	G	WO	100	0:15	0:15	0:01.74	7:11.9956	0:00	0:00	0014	D	3
15-Jul-2019 11:34:23	15-Jul-2019 11:55:32	28469	0001	010855	20		WO	15	0:15	0:15	0:00.41	8:54.6863	0:00	0:00	0001		
15-Jul-2019 11:55:32	15-Jul-2019 13:42:08	23557	0001	010954	20		WO	40	0:15	0:15	0:01.54	3:51.4629	0:00	0:00	0001		
15-Jul-2019 13:42:08	15-Jul-2019 15:13:44	28470	0001	010954	20		WO	40	0:15	0:15	0:01.54	6:31.0229	0:00	0:00	0001		
15-Jul-2019 15:13:44	16-Jul-2019 12:37:44	28380	0019	014221	20	G	WO	100	0:15	0:15	0:02.43	7:25.7495	0:00	0:00	0019	D	3
16-Jul-2019 12:37:44	17-Jul-2019 12:39:24	28617	0001	003433	20	G	WO	500	0:15	0:15	0:00.7	3:24.8043	0:00	0:00	0001	M	D
17-Jul-2019 12:39:24	17-Jul-2019 14:27:44	28380	0002	003433	20	G	WO	100	0:15	0:15	0:00.7	4:38.8043	0:00	0:00	0002	D	1,5
15-Aug-2019 11:07:45	19-Aug-2019 09:05:39	28487	0001	016231	20		WO	285	0:15	0:15	0:02.94	0:00	0:00	0:00	0001		
19-Aug-2019 09:05:39	19-Aug-2019 11:22:19	28380	0003	003434	20	G	WO	100	0:15	0:15	0:00.8	1 / 18:55.5	0:00	0:00	0003	D	1,5
19-Aug-2019 11:22:19	20-Aug-2019 16:22:19	28384	0001	015937	20		WO	360	0:15	0:15	0:02	2:07.5667	0:00	0:00	0001		
20-Aug-2019 16:22:19	21-Aug-2019 08:39:59	28380	0009	014526	20	G	WO	100	0:15	0:15	0:00.47	2:57.6971	0:00	0:00	0009	D	2
21-Aug-2019 08:39:59	21-Aug-2019 11:56:39	28380	0010	003445	20	G	WO	100	0:15	0:15	0:01.25	7:02.6997	0:00	0:00	0010	D	2
21-Aug-2019 11:56:39	22-Aug-2019 12:41:39	28384	0002	015937	20		WO	240	0:15	0:15	0:02	4:31.2997	0:00	0:00	0002		
22-Aug-2019 12:41:39	23-Aug-2019 09:19:39	28380	0022	004002	20	G	WO	100	0:15	0:15	0:02.31	4:46.0517	0:00	0:00	0022	D	3
23-Aug-2019 09:19:39	23-Aug-2019 10:09:39	27807	0001	012467	30		WO	20	0:15	0:15	0:01	7:31.1639	0:00	0:00	0001		
23-Aug-2019 10:09:39	27-Aug-2019 15:19:39	27710	0001	040843	45		WO	440	0:15	0:15	0:02.75	8:52.1639	0:00	0:00	0001		
27-Aug-2019 15:19:39	28-Aug-2019 13:05:37	27617	0001	031222	20		WO	120	0:00	0:00	0:03	3:31.1639	0:00	0:00	0001		
28-Aug-2019 13:05:37	28-Aug-2019 14:10:37	27930	0001	027657	30		WO	500	0:15	0:15	0:00.1	3:42.1239	0:00	0:00	0001		
28-Aug-2019 14:10:37	29-Aug-2019 11:20:37	28357	0001	037710	45		WO	100	0:15	0:15	0:03.25	2:12.1239	0:00	0:00	0001		
29-Aug-2019 11:20:37	30-Aug-2019 07:57:49	25530	0001	005069	20		WO	120	0:15	0:15	0:02.31	4:58.7239	0:00	0:00	0001		
30-Aug-2019 07:57:49	30-Aug-2019 11:55:49	28792	0002	028976	30		WO	208	0:15	0:15	0:01	7:18.5239	0:00	0:00	0002		
30-Aug-2019 11:55:49	30-Aug-2019 16:15:49	28792	0001	028976	30		WO	200	0:15	0:15	0:01	8:46.5239	0:00	0:00	0001		
30-Aug-2019 16:15:49	03-Sep-2019 15:12:49	28514	0001	005071	30		WO	350	0:15	0:15	0:02.52	6:16.5239	0:00	0:00	0001		
03-Sep-2019 15:12:49	03-Sep-2019 16:12:49	28954	0001	031623	30		WO	10	0:30	0:30	0:03	4:55.5239	0:00	0:00	0001		
03-Sep-2019 16:12:49	04-Sep-2019 10:15:19	28955	0001	031920	30		WO	50	0:30	0:30	0:02.75	6:30.5239	0:00	0:00	0001		
04-Sep-2019 10:15:19	04-Sep-2019 14:19:31	28931	0001	016525	20		WO	80	0:15	0:15	0:02.49	2:48.0239	0:00	0:00	0001		
04-Sep-2019 14:19:31	05-Sep-2019 09:24:31	28930	0001	016524	60		WO	20	0:30	0:30	0:10	2:22.2239	0:00	0:00	0001		
05-Sep-2019 09:24:31	05-Sep-2019 15:33:31	28923	0001	003606	20		WO	100	0:15	0:15	0:02.94	7:12.5464	0:00	0:00	0001		
05-Sep-2019 15:33:31	06-Sep-2019 08:52:31	28380	0008	003443	20	G	WO	100	0:15	0:15	0:00.93	7:47.1475	0:00	0:00	0008	D	2

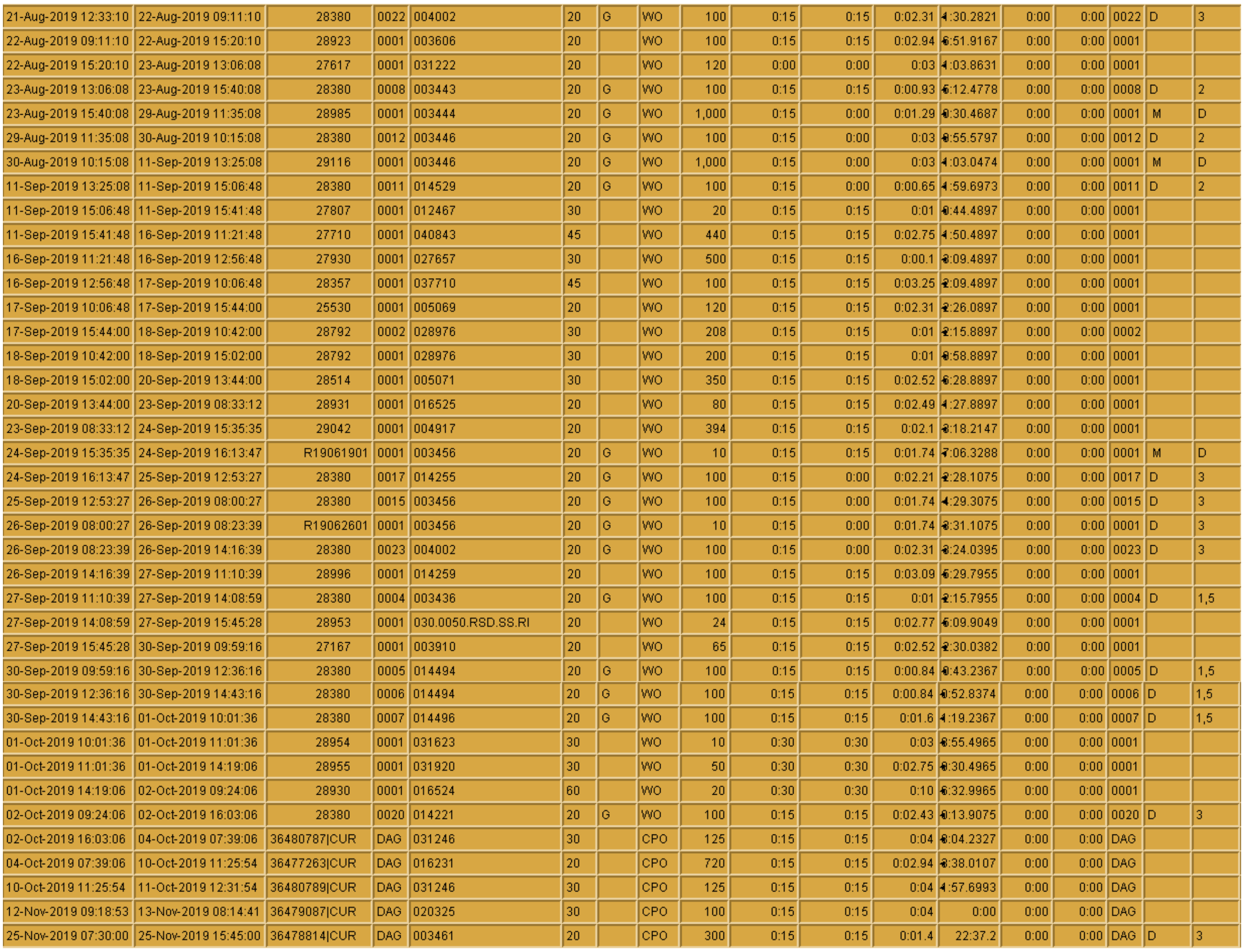


06-Sep-2019 08:52:31	11-Sep-2019 14:32:31	28985	0001	003444	20	G	WO	1,000	0:15	0:15	0:01.29	7:50.1384	0:00	0:00	0001	M	D
11-Sep-2019 14:32:31	13-Sep-2019 12:34:55	29042	0001	004917	20		WO	394	0:15	0:15	0:02.1	0:36.3489	0:00	0:00	0001		
13-Sep-2019 12:34:55	13-Sep-2019 13:13:07	R19061901	0001	003456	20	G	WO	10	0:15	0:15	0:01.74	4:12.9318	0:00	0:00	0001	M	D
13-Sep-2019 13:13:07	16-Sep-2019 10:07:07	28996	0001	014259	20		WO	100	0:15	0:15	0:03.09	4:33.5319	0:00	0:00	0001		
16-Sep-2019 10:07:07	16-Sep-2019 13:05:27	28380	0004	003436	20	G	WO	100	0:15	0:15	0:01	4:19.5319	0:00	0:00	0004	D	1,5
16-Sep-2019 13:05:27	16-Sep-2019 14:26:55	28953	0001	030.0050.RSD.SS.RI	20		WO	24	0:15	0:15	0:02.77	4:13.6412	0:00	0:00	0001		
16-Sep-2019 14:26:55	17-Sep-2019 13:21:55	28380	0012	003446	20	G	WO	100	0:15	0:15	0:03	2:54.6561	0:00	0:00	0012	D	2
17-Sep-2019 13:21:55	30-Sep-2019 07:31:55	29116	0001	003446	20	G	WO	1,000	0:15	0:15	0:03	0:17.1239	0:00	0:00	0001	M	D
30-Sep-2019 07:31:55	30-Sep-2019 13:26:35	28380	0017	014255	20	G	WO	100	0:15	0:15	0:02.21	7:53.5239	0:00	0:00	0017	D	3
30-Sep-2019 13:26:35	01-Oct-2019 08:48:35	28380	0015	003456	20	G	WO	100	0:15	0:15	0:01.74	6:09.7239	0:00	0:00	0015	D	3
01-Oct-2019 08:48:35	01-Oct-2019 12:32:23	27167	0001	003910	20		WO	65	0:15	0:15	0:02.52	8:40.8412	0:00	0:00	0001		
01-Oct-2019 12:32:23	01-Oct-2019 14:14:03	28380	0011	014529	20	G	WO	100	0:15	0:15	0:00.65	4:14.6404	0:00	0:00	0011	D	2
01-Oct-2019 14:14:03	02-Oct-2019 07:36:03	28380	0005	014494	20	G	WO	100	0:15	0:15	0:00.84	16:28.064	0:00	0:00	0005	D	1,5
02-Oct-2019 07:36:03	02-Oct-2019 09:58:03	28380	0006	014494	20	G	WO	100	0:15	0:15	0:00.84	6:00.3071	0:00	0:00	0006	D	1,5
02-Oct-2019 09:58:03	02-Oct-2019 14:16:23	28380	0007	014496	20	G	WO	100	0:15	0:15	0:01.6	12:04.064	0:00	0:00	0007	D	1,5
02-Oct-2019 14:16:23	03-Oct-2019 11:09:23	28380	0023	004002	20	G	WO	100	0:15	0:15	0:02.31	6:24.4559	0:00	0:00	0023	D	3
03-Oct-2019 11:09:23	04-Oct-2019 08:33:23	28380	0020	014221	20	G	WO	100	0:15	0:15	0:02.43	2:41.5972	0:00	0:00	0020	D	3
04-Oct-2019 08:33:23	07-Oct-2019 09:09:23	36480787|CUR	DAG	031246	30		CPO	125	0:15	0:15	0:04	8:01.5223	0:00	0:00	DAG		
07-Oct-2019 09:09:23	11-Oct-2019 13:26:11	36477263|CUR	DAG	016231	20		CPO	720	0:15	0:15	0:02.94	0:25.8204	0:00	0:00	DAG		
11-Oct-2019 13:26:11	14-Oct-2019 14:02:11	36480789|CUR	DAG	031246	30		CPO	125	0:15	0:15	0:04	13:57.989	0:00	0:00	DAG		
12-Nov-2019 09:18:53	13-Nov-2019 08:14:41	36479087|CUR	DAG	020325	30		CPO	100	0:15	0:15	0:04	0:00	0:00	0:00	DAG		
25-Nov-2019 07:30:00	25-Nov-2019 15:45:00	36478814|CUR	DAG	003461	20		CPO	300	0:15	0:15	0:01.4	22:37.2	0:00	0:00	DAG	D	3

Figure B1: Without Set-up Optimization



FP Start Date	FP End Date	Order No.	Line	Item	Op./ Split	Fixed	Order Type	Sched. Qty.	Setup Time [h:m]	Calculated Setup [h:m]	Run Time [h:m]	QT Mach. [h:m]	QT Mat. [h:m]	QT Tool [h:m]	Line ID	Attrib.1 Code	Attrib.1 Value
26-Jun-2019 12:30:00	27-Jun-2019 08:35:00	21751	0001	016535	30		WO	110	0:15	0:15	0:02.5	0:03	0:00	0:00	0001		
27-Jun-2019 08:35:00	28-Jun-2019 14:25:34	27478	0001	016231	20		WO	350	0:00	0:00	0:02.2445	20:08	0:00	0:00	0001		
28-Jun-2019 14:25:34	01-Jul-2019 15:16:34	27243	0001	031246	30		WO	125	0:15	0:15	0:04	4d / 0:53.7	0:00	0:00	0001		
01-Jul-2019 15:16:34	02-Jul-2019 10:27:40	27615	0001	STES.3200RSDSSRIACCI	20		WO	60	0:15	0:15	0:03.685	12:49.575	0:00	0:00	0001	D	3
02-Jul-2019 10:27:40	03-Jul-2019 07:36:40	28380	0019	014221	20	G	WO	100	0:15	0:00	0:02.43	22:00.675	0:00	0:00	0019	D	3
03-Jul-2019 07:36:40	04-Jul-2019 09:01:40	27749	0001	016535	30		WO	220	0:15	0:15	0:02.5	21:18.675	0:00	0:00	0001		
04-Jul-2019 09:01:40	08-Jul-2019 10:59:20	27026	0001	030.0100.RSDE.SS.ACC	20	G	WO	250	1:00	1:00	0:03	20:34.675	0:00	0:00	0001		
08-Jul-2019 10:59:20	08-Jul-2019 11:31:50	28266	0001	002367	20		WO	35	0:15	0:15	0:00.5	8:51.8417	0:00	0:00	0001		
08-Jul-2019 11:31:50	08-Jul-2019 16:17:10	28380	0018	014256	20	G	WO	100	0:15	0:15	0:01.69	2:15.1942	0:00	0:00	0018	D	3
08-Jul-2019 16:17:10	09-Jul-2019 11:57:10	28290	0001	014223	20		WO	100	0:15	0:00	0:02.65	6:00.5609	0:00	0:00	0001	D	3
09-Jul-2019 11:57:10	09-Jul-2019 13:59:52	28319	0001	014225	20		WO	30	0:15	0:00	0:03.09	4:35.2809	0:00	0:00	0001	D	3
09-Jul-2019 13:59:52	10-Jul-2019 10:09:52	28380	0021	004001	20	G	WO	100	0:15	0:00	0:02.1	6:47.0579	0:00	0:00	0021	D	3
10-Jul-2019 10:09:52	10-Jul-2019 10:59:52	28803	0001	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	20:57.256	0:00	0:00	0001		
10-Jul-2019 10:59:52	10-Jul-2019 11:49:52	28803	0002	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	8:43.3757	0:00	0:00	0002		
10-Jul-2019 11:49:52	10-Jul-2019 13:09:52	28803	0003	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	8:29.8557	0:00	0:00	0003		
10-Jul-2019 13:09:52	10-Jul-2019 13:59:52	28803	0004	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	0:46.3357	0:00	0:00	0004		
10-Jul-2019 13:59:52	10-Jul-2019 15:04:52	28803	0005	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	4:32.8157	0:00	0:00	0005		
10-Jul-2019 15:04:52	10-Jul-2019 15:54:52	28803	0006	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	7:34.2957	0:00	0:00	0006		
10-Jul-2019 15:54:52	11-Jul-2019 07:44:52	28803	0008	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	8:20.7757	0:00	0:00	0008		
11-Jul-2019 07:44:52	11-Jul-2019 08:34:52	28803	0009	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	0:07.2557	0:00	0:00	0009		
11-Jul-2019 08:34:52	11-Jul-2019 09:24:52	28803	0010	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	0:53.7357	0:00	0:00	0010		
11-Jul-2019 09:24:52	11-Jul-2019 10:29:52	28803	0011	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	4:40.2157	0:00	0:00	0011		
11-Jul-2019 10:29:52	11-Jul-2019 11:19:52	28803	0012	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	2:41.6957	0:00	0:00	0012		
11-Jul-2019 11:19:52	11-Jul-2019 12:39:52	28803	0013	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	8:28.1757	0:00	0:00	0013		
11-Jul-2019 12:39:52	11-Jul-2019 13:29:52	28803	0014	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	4:44.6557	0:00	0:00	0014		
11-Jul-2019 13:29:52	12-Jul-2019 08:31:33	28694	0001	006663	20		WO	84	0:15	0:15	0:02.52	6:02.2717	0:00	0:00	0001		
12-Jul-2019 08:31:33	12-Jul-2019 12:31:43	28693	0001	006662	20		WO	66	0:15	0:15	0:02.73	8:52.3357	0:00	0:00	0001		
12-Jul-2019 12:31:43	12-Jul-2019 13:21:43	28803	0007	030.0250.RSD.SF	20		WO	20	0:15	0:15	0:01.75	8:48.9957	0:00	0:00	0007		
12-Jul-2019 13:21:43	15-Jul-2019 08:43:43	28380	0014	003456	20	G	WO	100	0:15	0:15	0:01.74	4:46.1689	0:00	0:00	0014	D	3
15-Jul-2019 08:43:43	15-Jul-2019 09:04:52	28469	0001	010855	20		WO	15	0:15	0:15	0:00.41	8:23.1529	0:00	0:00	0001		
15-Jul-2019 09:04:52	15-Jul-2019 10:36:28	23557	0001	010954	20		WO	40	0:15	0:15	0:01.54	7:04.9296	0:00	0:00	0001		
15-Jul-2019 10:36:28	15-Jul-2019 11:53:04	28470	0001	010954	20		WO	40	0:15	0:15	0:01.54	8:29.4896	0:00	0:00	0001		
15-Jul-2019 11:53:04	16-Jul-2019 11:54:44	28617	0001	003433	20	G	WO	500	0:15	0:15	0:00.7	4:56.4176	0:00	0:00	0001	M	D
16-Jul-2019 11:54:44	16-Jul-2019 14:13:04	28380	0002	003433	20	G	WO	100	0:15	0:15	0:00.7	0:25.4176	0:00	0:00	0002	D	1,5
16-Jul-2019 14:13:04	16-Jul-2019 16:21:44	28380	0013	014531	20	G	WO	100	0:15	0:15	0:00.74	6:36.5109	0:00	0:00	0013	D	2
18-Jul-2019 09:05:36	18-Jul-2019 10:00:12	28406	0001	008138	20		WO	60	0:15	0:15	0:00.41	0:00	0:00	0:00	0001		
19-Jul-2019 12:53:36	12-Aug-2019 11:38:36	28512	0001	003465	20		WO	100	0:15	0:15	0:04.2	0:00	0:00	0:00	0001		
12-Aug-2019 11:38:36	13-Aug-2019 09:09:36	28482	0001	014226	20		WO	100	0:15	0:15	0:03.31	0:27.6037	0:00	0:00	0001		
13-Aug-2019 09:09:36	13-Aug-2019 14:34:36	28481	0001	014223	20		WO	100	0:15	0:15	0:02.65	8d / 17:41	0:00	0:00	0001		
13-Aug-2019 14:34:36	14-Aug-2019 11:14:16	28380	0016	014255	20	G	WO	100	0:15	0:15	0:02.21	2:48.4037	0:00	0:00	0016	D	3
14-Aug-2019 11:14:16	16-Aug-2019 09:12:10	28487	0001	016231	20		WO	285	0:15	0:15	0:02.94	0:19.8303	0:00	0:00	0001		
16-Aug-2019 09:12:10	16-Aug-2019 11:28:50	28380	0003	003434	20	G	WO	100	0:15	0:15	0:00.8	0:15.3303	0:00	0:00	0003	D	1,5
16-Aug-2019 11:28:50	19-Aug-2019 16:28:50	28384	0001	015937	20		WO	360	0:15	0:15	0:02	2:27.397	0:00	0:00	0001		
19-Aug-2019 16:28:50	20-Aug-2019 08:46:30	28380	0009	014526	20	G	WO	100	0:15	0:15	0:00.47	7:41.9274	0:00	0:00	0009	D	2
20-Aug-2019 08:46:30	20-Aug-2019 11:48:10	28380	0010	003445	20	G	WO	100	0:15	0:00	0:01.25	4:46.9301	0:00	0:00	0010	D	2
20-Aug-2019 11:48:10	21-Aug-2019 12:33:10	28384	0002	015937	20		WO	240	0:15	0:15	0:02	4:45.5301	0:00	0:00	0002		



21-Aug-2019 12:33:10	22-Aug-2019 09:11:10	28380	0022	004002	20	G	WO	100	0:15	0:15	0:02.31	4:30.2821	0:00	0:00	0022	D	3
22-Aug-2019 09:11:10	22-Aug-2019 15:20:10	28923	0001	003606	20		WO	100	0:15	0:15	0:02.94	6:51.9167	0:00	0:00	0001		
22-Aug-2019 15:20:10	23-Aug-2019 13:06:08	27617	0001	031222	20		WO	120	0:00	0:00	0:03	4:03.8631	0:00	0:00	0001		
23-Aug-2019 13:06:08	23-Aug-2019 15:40:08	28380	0008	003443	20	G	WO	100	0:15	0:15	0:00.93	6:12.4778	0:00	0:00	0008	D	2
23-Aug-2019 15:40:08	29-Aug-2019 11:35:08	28985	0001	003444	20	G	WO	1,000	0:15	0:00	0:01.29	0:30.4687	0:00	0:00	0001	M	D
29-Aug-2019 11:35:08	30-Aug-2019 10:15:08	28380	0012	003446	20	G	WO	100	0:15	0:00	0:03	8:55.5797	0:00	0:00	0012	D	2
30-Aug-2019 10:15:08	11-Sep-2019 13:25:08	29116	0001	003446	20	G	WO	1,000	0:15	0:00	0:03	4:03.0474	0:00	0:00	0001	M	D
11-Sep-2019 13:25:08	11-Sep-2019 15:06:48	28380	0011	014529	20	G	WO	100	0:15	0:00	0:00.65	4:59.6973	0:00	0:00	0011	D	2
11-Sep-2019 15:06:48	11-Sep-2019 15:41:48	27807	0001	012467	30		WO	20	0:15	0:15	0:01	0:44.4897	0:00	0:00	0001		
11-Sep-2019 15:41:48	16-Sep-2019 11:21:48	27710	0001	040843	45		WO	440	0:15	0:15	0:02.75	4:50.4897	0:00	0:00	0001		
16-Sep-2019 11:21:48	16-Sep-2019 12:56:48	27930	0001	027657	30		WO	500	0:15	0:15	0:00.1	3:09.4897	0:00	0:00	0001		
16-Sep-2019 12:56:48	17-Sep-2019 10:06:48	28357	0001	037710	45		WO	100	0:15	0:15	0:03.25	2:09.4897	0:00	0:00	0001		
17-Sep-2019 10:06:48	17-Sep-2019 15:44:00	25530	0001	005069	20		WO	120	0:15	0:15	0:02.31	2:26.0897	0:00	0:00	0001		
17-Sep-2019 15:44:00	18-Sep-2019 10:42:00	28792	0002	028976	30		WO	208	0:15	0:15	0:01	2:15.8897	0:00	0:00	0002		
18-Sep-2019 10:42:00	18-Sep-2019 15:02:00	28792	0001	028976	30		WO	200	0:15	0:15	0:01	8:58.8897	0:00	0:00	0001		
18-Sep-2019 15:02:00	20-Sep-2019 13:44:00	28514	0001	005071	30		WO	350	0:15	0:15	0:02.52	6:28.8897	0:00	0:00	0001		
20-Sep-2019 13:44:00	23-Sep-2019 08:33:12	28931	0001	016525	20		WO	80	0:15	0:15	0:02.49	4:27.8897	0:00	0:00	0001		
23-Sep-2019 08:33:12	24-Sep-2019 15:35:35	29042	0001	004917	20		WO	394	0:15	0:15	0:02.1	8:18.2147	0:00	0:00	0001		
24-Sep-2019 15:35:35	24-Sep-2019 16:13:47	R19061901	0001	003456	20	G	WO	10	0:15	0:15	0:01.74	7:06.3288	0:00	0:00	0001	M	D
24-Sep-2019 16:13:47	25-Sep-2019 12:53:27	28380	0017	014255	20	G	WO	100	0:15	0:00	0:02.21	2:28.1075	0:00	0:00	0017	D	3
25-Sep-2019 12:53:27	26-Sep-2019 08:00:27	28380	0015	003456	20	G	WO	100	0:15	0:00	0:01.74	4:29.3075	0:00	0:00	0015	D	3
26-Sep-2019 08:00:27	26-Sep-2019 08:23:39	R19062601	0001	003456	20	G	WO	10	0:15	0:00	0:01.74	8:31.1075	0:00	0:00	0001	D	3
26-Sep-2019 08:23:39	26-Sep-2019 14:16:39	28380	0023	004002	20	G	WO	100	0:15	0:00	0:02.31	8:24.0395	0:00	0:00	0023	D	3
26-Sep-2019 14:16:39	27-Sep-2019 11:10:39	28996	0001	014259	20		WO	100	0:15	0:15	0:03.09	6:29.7955	0:00	0:00	0001		
27-Sep-2019 11:10:39	27-Sep-2019 14:08:59	28380	0004	003436	20	G	WO	100	0:15	0:15	0:01	2:15.7955	0:00	0:00	0004	D	1,5
27-Sep-2019 14:08:59	27-Sep-2019 15:45:28	28953	0001	030.0050.RSD.SS.RI	20		WO	24	0:15	0:15	0:02.77	6:09.9049	0:00	0:00	0001		
27-Sep-2019 15:45:28	30-Sep-2019 09:59:16	27167	0001	003910	20		WO	65	0:15	0:15	0:02.52	2:30.0382	0:00	0:00	0001		
30-Sep-2019 09:59:16	30-Sep-2019 12:36:16	28380	0005	014494	20	G	WO	100	0:15	0:15	0:00.84	0:43.2367	0:00	0:00	0005	D	1,5
30-Sep-2019 12:36:16	30-Sep-2019 14:43:16	28380	0006	014494	20	G	WO	100	0:15	0:15	0:00.84	0:52.8374	0:00	0:00	0006	D	1,5
30-Sep-2019 14:43:16	01-Oct-2019 10:01:36	28380	0007	014496	20	G	WO	100	0:15	0:15	0:01.6	4:19.2367	0:00	0:00	0007	D	1,5
01-Oct-2019 10:01:36	01-Oct-2019 11:01:36	28954	0001	031623	30		WO	10	0:30	0:30	0:03	8:55.4965	0:00	0:00	0001		
01-Oct-2019 11:01:36	01-Oct-2019 14:19:06	28955	0001	031920	30		WO	50	0:30	0:30	0:02.75	8:30.4965	0:00	0:00	0001		
01-Oct-2019 14:19:06	02-Oct-2019 09:24:06	28930	0001	016524	60		WO	20	0:30	0:30	0:10	6:32.9965	0:00	0:00	0001		
02-Oct-2019 09:24:06	02-Oct-2019 16:03:06	28380	0020	014221	20	G	WO	100	0:15	0:15	0:02.43	0:13.9075	0:00	0:00	0020	D	3
02-Oct-2019 16:03:06	04-Oct-2019 07:39:06	36480787|CUR	DAG	031246	30		CPO	125	0:15	0:15	0:04	8:04.2327	0:00	0:00	DAG		
04-Oct-2019 07:39:06	10-Oct-2019 11:25:54	36477263|CUR	DAG	016231	20		CPO	720	0:15	0:15	0:02.94	8:38.0107	0:00	0:00	DAG		
10-Oct-2019 11:25:54	11-Oct-2019 12:31:54	36480789|CUR	DAG	031246	30		CPO	125	0:15	0:15	0:04	4:57.6993	0:00	0:00	DAG		
12-Nov-2019 09:18:53	13-Nov-2019 08:14:41	36479087|CUR	DAG	020325	30		CPO	100	0:15	0:15	0:04	0:00	0:00	0:00	DAG		
25-Nov-2019 07:30:00	25-Nov-2019 15:45:00	36478814|CUR	DAG	003461	20		CPO	300	0:15	0:15	0:01.4	22:37.2	0:00	0:00	DAG	D	3

Figure B2: With Set-up Optimization

Appendix C: Pseudo Code

Algorithm 1: Genetic Algorithm – Variable Neighborhood Search

```
Load input data of the jobs to be scheduled
Set algorithm parameters
For Individual = 1 to Population Size do
    Create an initial schedule
    Calculate fitness of this schedule                      » Algorithm 2: Fitness Calculation
Next Individual
For Generation = 1 to Number of Generations do
    Sort all individuals based on the fitness value
    Copy the best individuals to the Next Generation Population    » Elite preservation
    For Individual = 1 to Population Size do
        Tournament Selection                                » Algorithm 3: Tournament Selection
        Copy selected individual to Offspring Population
    Next Individual
    For Individual = Elite Size* to Population Size do
        If Random Number ≤ Crossover Probability then
            Apply crossover to this individual              » Algorithm 4: Crossover
        End if
        If Random Number ≤ Mutation Probability then
            Apply mutation to this individual               » Algorithm 5: Mutation
        End if
        Copy this individual to the Next Generation Population
    Next Individual
    Sort the population based on fitness value
    Variable Neighborhood Search                            » Algorithm 6: VNS
Next Generation
```

* *Elite Size = Elite Reserve Ratio * Population Size*

Algorithm 2: Fitness Calculation

```
For Operation = 1 to TotalOperations do
    StartTime = Max(JobReleaseDate; MachineReleaseDate; WorkerReleaseDate)
    JobReleaseDate = StartTime + ProcessingTime/MachineCapacity
    MachineReleaseDate = StartTime + ProcessingTime/MachineCapacity
    WorkerReleaseDate = StartTime + ProcessingTime/MachineCapacity
    NumberOfOperationsScheduled = NumberOfOperationsScheduled + 1
    If NumberOfOperationsScheduled = NumberOfOperations then
        JobLateness = DueDate – JobReleaseDate
        Lateness = Lateness + JobLateness
    End if
Next Operation
Return Lateness
```

Algorithm 3: Tournament Selection

```
For TournamentIndividual = 1 to TournamentSize do
    Randomly select an individual
Next TournamentIndividual
Sort the TournamenIndividuals based on the fitness values
Return Individual with the best fitness value
```

Algorithm 4: Crossover

```
Randomly select two different genes
For Offspring = 1 to 2 do
    For Gene = 1 to TotalGenes do
            If RandomNumber ≤ 0.50 then
                    Copy this gene of parent1 to offspring1
            Else
                    Find the operation on this gene of parent1 on parent2
                    Remember the gene of parent2
            End if
    Next
    Fill the empty genes of offspring1 with the missing operations in the same sequence as in which they are scheduled on parent 2
    Check the feasibility of the offspring schedule
    Calculate Fitness of this schedule               » Algorithm 2: Fitness Calculation
Next
If fitness of offspring1 < fitness of offspring2 then
    Return offspring1
Else
    Return offspring2
End if
```

Algorithm 5: Mutation

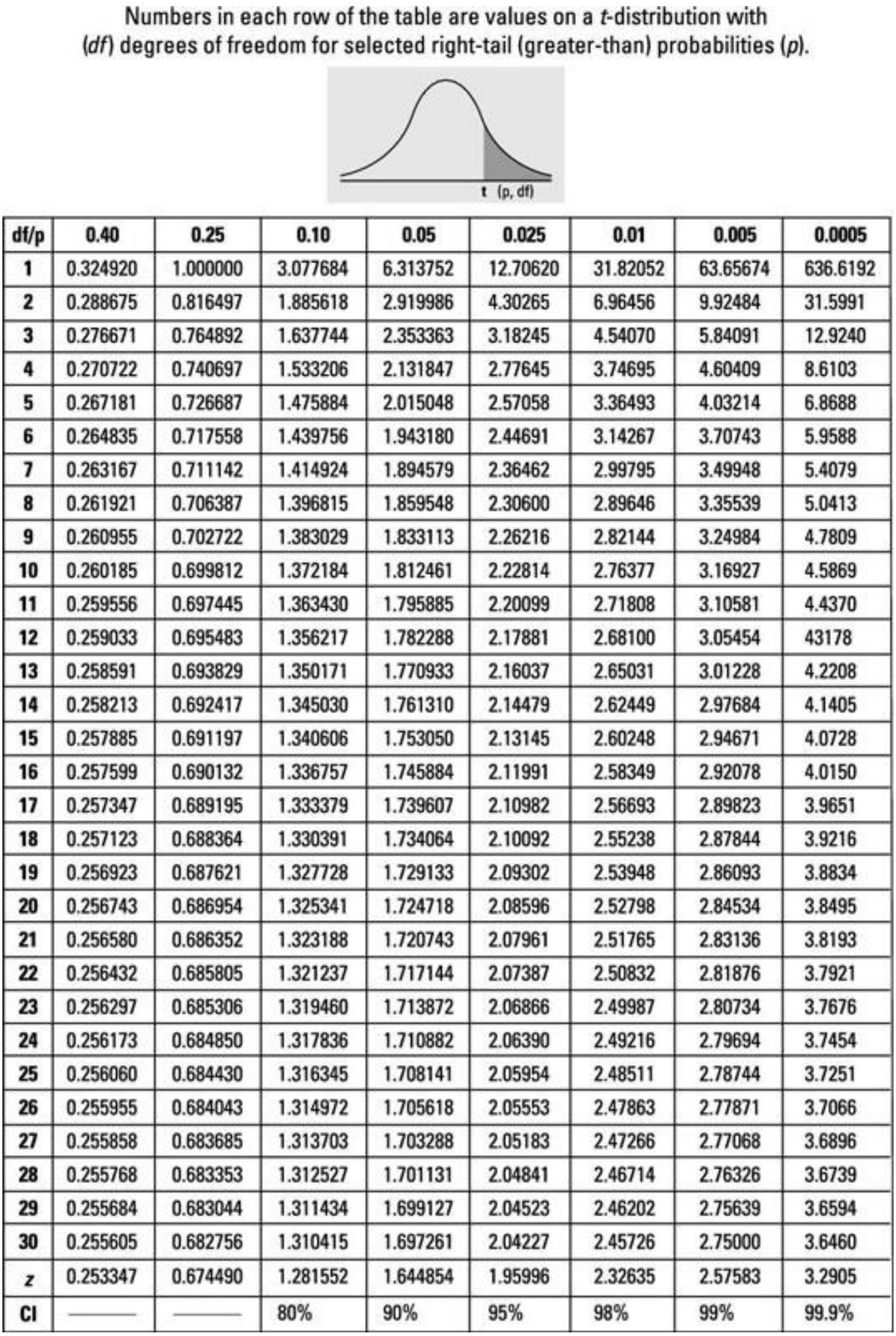
```
Randomly select two different genes
Swap genes
Check the feasibility of the mutated individual
Randomly select a gene
Replace the machine by a random machine of the same machine-group
If  the worker of this gene does not belong to the machine-cluster of the machine then
    Randomly pick a worker from the machine-cluster of the machine
End if
Randomly select a gene
Check the machine-cluster to which the machine at this gene belongs
Randomly select a worker that belongs to the same machine-cluster
Return mutated individual
```

Algorithm 6: VNS

```
For Individual = 1 to VNSPopulation*   do
    Create a copy of this Individual, the Neighbor
    A = 0
    B = 0
    C = 0
    D = 0
    For iteration = 1 to TotalVNSIterations do
        If A = 0 OR (A = 1 AND B = 1 AND C = 1) then
            Randomly select two different genes of the neighbor schedule
            Swap genes
            Check the feasibility of the mutated neighbor
            If fitness of initial individual ≥ fitness of mutated neighbor then
                Copy the schedule of the neighbor to the initial individual
                A = 0
                D = 0
            Else
                Copy the initial individual to the schedule of the neighbor
                A = 1 – A
                D = D + 1
            End if
        End if
        If B = 0  then
            Randomly select a gene of the neighbor schedule
            Replace the machine by a random machine of the same machine
            group
            If  the worker of this gene does not belong to the machine-cluster of
            the machine then
                Randomly pick a worker from the machine-cluster of the
                machine
            End if
            If fitness of initial individual ≥ fitness of mutated neighbor then
                Copy the schedule of the neighbor to the initial individual
                B = 0
                D = 0
            Else
                Copy the initial individual to the schedule of the neighbor
                B = 1 – B
                D = D + 1
            End if
        End if
        If C = 0  then
            Randomly select a gene of the neighbor schedule
            Check the machine-cluster to which the machine at this gene belongs
            Randomly select a worker that belongs to the same machine-cluster
            If fitness of initial individual ≥ fitness of mutated neighbor then
                Copy the schedule of the neighbor to the initial individual
                C = 0
                D = 0
            Else
                Copy the initial individual to the schedule of the neighbor
                C = 1 – C
                D = D + 1
            End if
        End if
        If  D > (TotalVNSIterations / 5) then
            Exit for
        End if
    Next iteration
    Calculate fitness of this individual              » Algorithm 2: Fitness Calculation
Next Individual
Return Population
```

**VNSPopulation = 0.8 * Population Size*

Appendix D: T-Distribution Table



Numbers in each row of the table are values on a *t*-distribution with (*df*) degrees of freedom for selected right-tail (greater-than) probabilities (*p*).

df/p	0.40	0.25	0.10	0.05	0.025	0.01	0.005	0.0005
1	0.324920	1.000000	3.077684	6.313752	12.70620	31.82052	63.65674	636.6192
2	0.288675	0.816497	1.885618	2.919986	4.30265	6.96456	9.92484	31.5991
3	0.276671	0.764892	1.637744	2.353363	3.18245	4.54070	5.84091	12.9240
4	0.270722	0.740697	1.533206	2.131847	2.77645	3.74695	4.60409	8.6103
5	0.267181	0.726687	1.475884	2.015048	2.57058	3.36493	4.03214	6.8688
6	0.264835	0.717558	1.439756	1.943180	2.44691	3.14267	3.70743	5.9588
7	0.263167	0.711142	1.414924	1.894579	2.36462	2.99795	3.49948	5.4079
8	0.261921	0.706387	1.396815	1.859548	2.30600	2.89646	3.35539	5.0413
9	0.260955	0.702722	1.383029	1.833113	2.26216	2.82144	3.24984	4.7809
10	0.260185	0.699812	1.372184	1.812461	2.22814	2.76377	3.16927	4.5869
11	0.259556	0.697445	1.363430	1.795885	2.20099	2.71808	3.10581	4.4370
12	0.259033	0.695483	1.356217	1.782288	2.17881	2.68100	3.05454	43178
13	0.258591	0.693829	1.350171	1.770933	2.16037	2.65031	3.01228	4.2208
14	0.258213	0.692417	1.345030	1.761310	2.14479	2.62449	2.97684	4.1405
15	0.257885	0.691197	1.340606	1.753050	2.13145	2.60248	2.94671	4.0728
16	0.257599	0.690132	1.336757	1.745884	2.11991	2.58349	2.92078	4.0150
17	0.257347	0.689195	1.333379	1.739607	2.10982	2.56693	2.89823	3.9651
18	0.257123	0.688364	1.330391	1.734064	2.10092	2.55238	2.87844	3.9216
19	0.256923	0.687621	1.327728	1.729133	2.09302	2.53948	2.86093	3.8834
20	0.256743	0.686954	1.325341	1.724718	2.08596	2.52798	2.84534	3.8495
21	0.256580	0.686352	1.323188	1.720743	2.07961	2.51765	2.83136	3.8193
22	0.256432	0.685805	1.321237	1.717144	2.07387	2.50832	2.81876	3.7921
23	0.256297	0.685306	1.319460	1.713872	2.06866	2.49987	2.80734	3.7676
24	0.256173	0.684850	1.317836	1.710882	2.06390	2.49216	2.79694	3.7454
25	0.256060	0.684430	1.316345	1.708141	2.05954	2.48511	2.78744	3.7251
26	0.255955	0.684043	1.314972	1.705618	2.05553	2.47863	2.77871	3.7066
27	0.255858	0.683685	1.313703	1.703288	2.05183	2.47266	2.77068	3.6896
28	0.255768	0.683353	1.312527	1.701131	2.04841	2.46714	2.76326	3.6739
29	0.255684	0.683044	1.311434	1.699127	2.04523	2.46202	2.75639	3.6594
30	0.255605	0.682756	1.310415	1.697261	2.04227	2.45726	2.75000	3.6460
z	0.253347	0.674490	1.281552	1.644854	1.95996	2.32635	2.57583	3.2905
CI	———	———	80%	90%	95%	98%	99%	99.9%

Figure from https://www.dummies.com/

Appendix E: Results

Table E1, E2 and E3 display the output results of the experiments described in Chapter 5. These are the average values for the 10 replications. Lateness1 is the value of the objective at the end of the algorithm but prior to any additional procedures such as post-processing and the algorithms that additionally run the SD. Lateness2 is the value of the objective after all of these additional procedures. For the experiments that do not include post processing nor SD as an additional procedure the values of Lateness1 and Lateness2 are the same.

Experiment Nr.	Run Time (sec.)	Lateness1 (days)	Lateness2 (days)	No. Jobs Late	OTDP (%)	ADL (days)	AQT (days)	AST (%)	Makespan (days)
1	46.97	-198.8	-198.8	4.2	72.0	-57.2	7.0	1.70	29.9
2	79.11	-214.6	-214.6	3.2	78.7	-68.5	5.9	1.70	30.2
3	53.99	-194.7	-194.7	3.5	76.7	-83.4	7.3	1.70	30.9
4	104.84	-215.2	-215.2	2.8	81.3	-67.0	5.9	1.70	30.9
5	34.43	-232.7	-232.7	2.9	80.7	-84.2	4.7	1.70	28.7
6	7.06	21.6	21.6	9.3	38.0	1.2	21.7	1.70	42.7
7	76.55	-194.7	-197.6	3.4	77.3	-86.4	7.1	1.70	30.9
8	115.28	-215.2	-216.9	2.8	81.3	-67.5	5.8	1.70	30.7
9	65.09	-194.0	-194.0	4.1	72.7	-71.0	7.2	2.75	30.7
10	103.49	-211.3	-211.3	3.3	78.0	-87.1	6.1	2.75	29.5
11	37.48	-228.5	-228.5	3.0	80.0	-97.2	4.9	2.74	29.1
12	18.62	-39.9	-39.9	7.9	47.3	-7.1	17.4	2.74	41.1
13	89.31	-194.0	-196.3	3.9	74.0	-82.4	7.0	2.75	30.5
14	124.79	-211.3	-212.4	3.3	78.0	-87.4	6.0	2.75	29.3
15	66.54	-112.5	-112.5	7.1	52.7	-17.4	12.8	2.76	36.2
16	101.69	-139.7	-139.7	5.3	64.7	-42.4	10.9	2.76	34.4
17	37.13	-153.4	-153.4	5.6	62.7	-36.3	10.0	2.76	34.4
18	24.44	152.2	152.2	11.4	24.0	12.6	30.4	2.76	53.9
19	72.71	-113.5	-113.5	6.8	54.7	-19.5	12.6	2.75	35.0
20	115.64	-124.9	-124.9	7.2	52.0	-18.8	11.8	2.75	33.9
21	39.22	-118.6	-118.6	7.2	52.0	-18.2	12.2	2.74	36.1
22	32.97	147.8	147.8	11.4	24.0	12.6	30.0	2.74	53.1
23	91.80	-112.5	-116.1	7.0	53.3	-18.1	12.5	2.76	36.0
24	130.63	-139.7	-141.6	5.3	64.7	-42.8	10.8	2.76	34.3
25	32.35	-113.5	-116.5	6.7	55.3	-20.0	12.4	2.75	34.6
26	48.35	-124.9	-128.0	7.1	52.7	-19.4	11.6	2.75	34.0
27	26.99	-114.9	-114.9	6.8	54.7	-18.1	12.6	2.76	35.3
28	41.55	-136.7	-136.7	5.0	66.7	-43.5	11.1	2.76	35.8
29	14.02	-157.0	-157.0	5.4	64.0	-36.5	9.8	2.76	34.0
30	8.31	150.7	150.7	11.4	24.0	12.4	30.3	2.76	53.9
31	27.10	-118.4	-118.4	6.7	55.3	-20.5	12.3	2.75	34.5
32	42.41	-132.6	-132.6	6.6	56.0	-26.1	11.4	2.75	33.1
33	14.35	-129.7	-129.7	6.6	56.0	-20.6	11.5	2.74	35.9
34	11.71	141.9	141.9	11.3	24.7	12.1	29.6	2.74	52.6
35	35.01	-114.9	-118.5	6.7	55.3	-18.8	12.3	2.76	35.0
36	46.80	-136.7	-138.2	4.9	67.3	-44.0	11.0	2.76	35.7

37	33.37	-118.4	-121.9	6.7	55.3	-21.1	12.0	2.75	34.2
38	49.45	-132.6	-136.3	6.5	56.7	-26.9	11.1	2.75	33.1

Table E1: Results Experiments Set 1

Experiment Nr.	Run Time (sec.)	Lateness1 (days)	Lateness2 (days)	No. Jobs Late	OTDP (%)	ADL (days)	AQT (days)	AST (%)	Makespan (days)
1	67.72	-194.7	0.7	8.8	41.3	-0.7	20.3	2.75	45.2
2	62.08	-215.2	-16.6	9.0	40.0	-10.4	19.1	2.75	44.4
3	13.26	-232.7	-28.2	8.7	42.0	-3.9	18.4	2.76	40.4
4	5.08	21.6	181.3	11.4	24.0	15.1	32.3	2.75	55.4
5	33.51	-194.7	0.8	8.9	40.7	-0.5	20.3	2.76	45.2
6	44.99	-215.2	-16.8	9.0	40.0	-10.4	19.1	2.75	44.3
7	25.68	-194.0	48.4	10.8	28.0	3.9	23.4	2.73	45.0
8	39.77	-211.3	30.8	9.7	35.3	1.8	22.2	2.73	44.6
9	13.48	-228.5	22.3	10.6	29.3	1.4	21.6	2.73	44.1
10	6.53	-39.9	183.9	11.7	22.0	15.3	32.4	2.74	56.0
11	32.50	-194.0	47.2	10.8	28.0	3.7	23.3	2.73	44.8
12	44.58	-211.3	30.6	9.7	35.3	1.7	22.2	2.73	44.5
13	26.51	-112.5	-112.5	7.1	52.7	-17.4	12.8	2.75	36.2
14	40.89	-139.7	-139.7	5.3	64.7	-42.4	10.9	2.74	34.4
15	13.70	-153.4	-153.5	5.6	62.7	-36.4	10.0	2.75	34.4
16	8.68	152.2	152.2	11.4	24.0	12.6	30.4	2.75	53.9
17	33.84	-112.5	-116.1	7.0	53.3	-18.1	12.5	2.75	36.0
18	46.66	-139.7	-141.6	5.3	64.7	-42.8	10.8	2.74	34.3
19	26.86	-114.9	-114.9	6.8	54.7	-18.1	12.6	2.74	35.3
20	41.28	-136.7	-136.7	5.0	66.7	-43.5	11.1	2.75	35.8
21	13.90	-157.0	-157.0	5.4	64.0	-36.5	9.8	2.75	34.0
22	8.25	150.7	150.7	11.4	24.0	12.4	30.3	2.75	53.9
23	34.81	-114.9	-118.5	6.7	55.3	-18.8	12.4	2.74	35.0
24	46.32	-136.7	-138.2	4.9	67.3	-44.0	11.0	2.75	35.7

Table E2: Results Experiments Set 2

Experiment Nr.	Run Time (sec.)	Lateness1 (days)	Lateness2 (days)	No. Jobs Late	OTDP (%)	ADL (days)	AQT (days)	AST (%)	Makespan (days)
1	1019.97	877.8	877.8	65.7	34.3	13.1	70.9	1.13	114.5
2	984.79	568.8	568.8	59	41	9.3	67.7	1.10	115.1
3	492.39	720.2	15070.6	99.9	0.1	150.8	212.8	1.11	285.2
4	494.5	728.5	14582.3	99.3	0.7	146.8	207.8	1.09	283.7

Table E3: Results Experiments Set 3