MASTER

Job scheduling in a quick response manufacturing environment
a bi-objective multi-resource hybrid flow shop with partial blocking

van Malsen, H.G.A.

Award date:
2020

Job Scheduling in a Quick Response Manufacturing Environment

—

A bi-objective multi-resource hybrid flow shop with partial blocking

H.G.A. (Renzo) van Malsen

In partial fulfillment of the requirements for the degree of
Master of Science
In Operations Management and Logistics

Supervisors
Dr. ir. N.P. Dellaert, Eindhoven University of Technology
Dr. T.G. Martagan, Eindhoven University of Technology
Ir. O. van Biezen, Royal Dekker

Eindhoven, March 2020

Abstract

Classical job and flow shops, with unlimited buffer capacity and large order sizes, have been extensively researched. Multiple solution methodologies have been explored, mainly approximation techniques, because of the complexity involved. However, underlying assumptions of those models no longer hold in today's highly flexible manufacturing environments. Demand nowadays is exposed to highly varying product portfolios and large series are becoming rare. This paper presents the implementation of multiple sophisticated scheduling techniques, such as a hybridized priority dispatching rule algorithm and a genetic algorithm, to minimize both total makespan and tardiness in a hybrid multi-resource flow shop with partial blocking constraints. As a case study, simulations have been performed at a wood processing facility subjected to a Quick Response Manufacturing (QRM) process. Besides, a two-stage flow model with limited buffer capacity has been analyzed by loosening some of the constraints issued by QRM. Our simulation study shows the effectiveness of a limited buffer with respect to tardiness minimization, whereas the QRM-based process performs best on minimization of the makespan.

Executive Summary

This report is the result of a study on job scheduling within a Quick Response Manufacturing (QRM) environment. As a case study, simulations have been performed at a wood processing facility of a Dutch manufacturer named Royal Dekker. Over the past decades, demand characteristics within the building industry have shifted, increasing the need to implement more flexible and cycle time focused scheduling strategies. As a result, Royal Dekker implemented QRM at one of its production facilities. Quick Response Manufacturing is a strategy for reducing lead times throughout the complete organization, mainly focusing on reducing idle time rather than production time (Suri, 2010). This involves, among other things, team-based manufacturing and job assignment to dedicated teams. That is, a team executes all production steps of a single order consecutively eventually resulting in no intermediate buffers.

Although QRM had significantly improved throughput times of production orders, it did not solve all problems encountered by Royal Dekker. These problems include low on-time delivery performance, high production costs per unit and in some cases a low throughput per unit of time. To overcome these burdens (i.e. improving performances) an in-depth analysis of process parameters is performed, which are then used to build multiple production configurations. These configurations are tested on actual data sets, showing significant improvements in terms of both makespan and total tardiness.

Team Composition

Firstly, an analysis was performed regarding team compositions on the shop floor. Currently, production teams are always composed of three employees. Based on the tasks a team needs to execute this team composition was chosen to be the most effective one by Royal Dekker. Next to the current composition, two alternative approaches were suggested during this research. One method, named Fixed-2/3, represents a mixed setting of teams with two and three operators respectively. Based on its characteristics a job is either classified as handleable by two or three operators, separating the demand into two lists based on required team size. Within the fixed composition teams are restricted to processing orders from the corresponding job list only. On the contrary, a variable method, named Variable-2/3, allows teams with three operators to process jobs intended for teams composed of two operators in case this order has the nearest due date.

After testing both methods on actual data sets, observed improvement for the makespan, in comparison with the current team setting, was 6.7% and 3.1% for the fixed and variable approach respectively. Tardiness is reduced by 54.8% when using the fixed team composition, whereas the variable methodology leads to a reduction of 32.3%. Both methods achieve improvements compared to the use of teams with three operators only. We conclude that by implementing the Fixed-2/3 methodology performances will be significantly improved.

Model Design

The actual shop floor at Royal Dekker can best be denoted as a multi-resource hybrid flow shop with partial blocking constraints. Figure 0.1 represents the flow line model of the shop floor. A job has to pass through at least one stage following a predetermined uni-directional production path. In order for a job to be processed both a machine and a team are required to execute an operation, turning the problem into a multi-resource optimization problem. One can talk about a flow shop being hybrid in case at least one stage has multiple (identical) machines. This is the case for two out of four production processes at Royal Dekker. Besides, partial blocking constraints apply since teams are dedicated to a single job and cannot start processing other jobs in case the required machine is not available due to another job being processed. This blocking constraint is indirectly related to the limited buffer capacity at the production facility.

Although not the main goal of this research, a Two-Stage flow model (see Figure 0.2) with limited buffer capacity is designed to evaluate the performances of the QRM-Based production process with performances obtained by the more general Two-Stage model. Cornerstone of this flow model is still the QRM philosophy but it allows a limited number of orders to be buffered and loosens the constraint that all operations of a single order have to be performed by a single team.

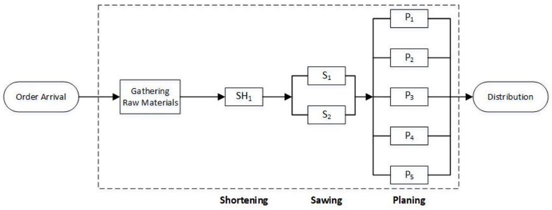

Figure 0.1: QRM-Based flow model

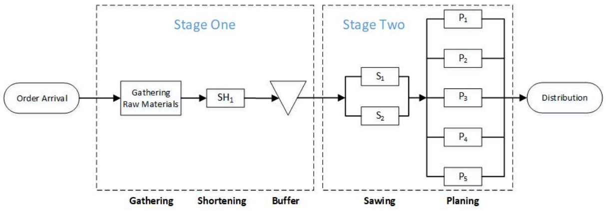

Figure 0.2: Two-Stage flow model

Both models were coded in the programming language of Excel, Microsoft Visual Basics for Applications (VBA). Despite VBA being far from an efficient programming language, it was chosen because of its direct connection to Excel, enabling Royal Dekker to use the models during day-to-day operations. After running multiple simulation runs, scheduling jobs using the EDD-SPT algorithm explained below, we conclude that the QRM-Based flow model performs best on minimizing makespan, whereas the Two-Stage flow model achieves by far the lowest total tardiness.

Scheduling Optimization

Furthermore, to schedule jobs more efficiently multiple sequencing heuristics are developed and reviewed. By evaluating the current sequencing procedure at Royal Dekker it could be determined an earliest due date technique is applied. Although, known to be easy to implement and computationally efficient, single priority dispatching rules, on the whole, deliver a bad job in minimizing the makespan. To solve the bi-objective problem introduced in this research a so-called hybrid algorithm is developed, in which both earliest due date (EDD) and shortest processing time (SPT) rules are used. The EDD-SPT algorithm prioritizes jobs firstly according to their due date and secondly to their processing time.

Besides a genetic algorithm (GA) is developed making use of roulette wheel parent selection and linear order crossover (LOX). Parameter fitting to our problem was applied by differentiating the crossover probability, mutation probability, number of elite parents, and the number of generations. Job sequences found by the EDD-SPT algorithm are used to form initial solutions to the GA.

Overall, the genetic algorithm leads to better results than both the EDD-rule and hybridized algorithm. It should, however, be mentioned that computation times of the GA rise to 16 minutes whereas the EDD-SPT algorithm finds slightly worse results in just 5 seconds.

Recommendations

Based on the simulation study of different QRM-Based configurations, we recommend Royal Dekker to switch to a more flexible team composition in case they want to stick to the QRM philosophy. Best would be to adopt the fixed composition of two teams with 3 operators and three teams with 2 operators. One should, however, take this method's sensitivity to the ratio of orders handleable by either 2 or 3 operators into account. In case the ratio changes beyond certain limits, it can be wise to review whether the Fixed-2/3 still performs best or one should switch to the Variable-2/3 approach.

Secondly, our research shows a great improvement in performances by using a genetic algorithm to sequence jobs. We recommend implementing this optimization heuristic in day-to-day scheduling. Implementation would be relatively easy as the programming language used is directly linked to Excel, which in turn connects with Royal Dekker's ERP-system.

Lastly, analysis of a Two-Stage flow model with a buffer capacity of 18m3 showed excellent results regarding total tardiness. With respect to total completion time, the Two-Stage requires slightly more labour to finish all jobs. However, compared to the significant reduction in tardy jobs our advise is to implement this production strategy. This, though, would include partially deviating from the currently used QRM philosophy.

Contents

List of Tables

List of Figures

1 Introduction

In an ever increasing competitive market, companies have to be even more efficient and flexible to survive. Over the past decades, effective scheduling has become a necessity to gain market-share and reduce costs. Companies failing to meet due dates committed to their customers will eventually experience decreasing customer satisfaction leading to losses as customers drop out. It is therefore, scheduling literature is rich. Due to the complexity of real world scheduling problems and the broad applicability in various industries it has become one of the most important issues in Operations Research (Wang et al., 2006).

One of those markets subjected to fierce competition is the wood processing industry. For over a century it has been a relatively conventional industry, overtaken by competing products such as plastics and steel. However, these changes in buyer preferences opened the eyes of many wood manufacturers now trying to make up for the past few years. New challenges are ahead, such as shortening delivery times. A key player in the Dutch and German market is Royal Dekker. They came to the understanding that changes were required to keep their market-share. To speed-up their production process, and hence realise short delivery times, Quick Response Manufacturing (QRM) was implemented throughout one of their facilities. Although their strategy did reduce the cycle time, other problems arose. For instance, throughput rates had decreased and as a result overall costs increased. These problems urge the need for Royal Dekker to schedule jobs more efficiently together with improving the throughput rate.

This study aims to improve the production efficiency of a single manufacturing facility within Royal Dekker. The problem can be characterized as a bi-objective multi-resource hybrid flow shop with partial blocking constraints. Besides the development of mathematical models and heuristics to fulfill the request of the commissioning company, this master thesis targets to narrow the gap between theory and practice. This is done by testing all models on realistically sized problems, while strictly limiting the number of assumptions made.

2 Problem Context

In order to get a clear and complete understanding of the context in which this thesis is executed, this chapter will provide a description of the company and some broad explanation of topics concerning the research area. As the research topic is directly related to problems faced by the production department the main focus of this chapter will be on the operations of this department rather than describing the company as a whole.

2.1 Company Description

Royal Dekker is a fully vertically integrated timber supplier with its own forest concessions, sawmills, factories, and assembly possibilities. Their main business consists of supplying timber products and (hard)wood in Europe to the building materials industry, manufacturing industry, construction industry, and retail. In order to do so Royal Dekker is in possession of a several warehouses with a combined floor space of 50.000 m^2, a distribution center of 22.000 m^2 and two production facilities with a total floor space of 24.000 m^2.

Founded in 1885, the company grew steadily and currently is one of the leading timber companies within Europe specialized in a broad range of products. In its early years the main focus was on trading wooden products, whereas during the last decades production of both stock keeping units and custom made products have taken a more distinctive position. As of today, almost 65% of Dekker's revenue is generated by importing wood and selling it directly to its customer, while the remaining 35% is generated by processing (i.e. plane, saw or assemble) timber products.

The head office of Royal Dekker is located in The Hague, a city on the western coast of the Netherlands. For many years The Hague has been the major production location, housing both a production and a wood coating facility. In 2015 a large reorganization of the production department took place which among other things involved a partial closure of the production facility and shutting down the coating facility located in The Hague. Most of the production activities (including coating) now take place within Vianen, a small city in the central Netherlands. After the reorganization Royal Dekker started implementing Quick Response Manufacturing (QRM) within its smallest facility (The Hague), which would have been implemented throughout the complete department in case the result was positive. Although QRM structured the process at the shop floor and led to positive results in terms of lead times, it negatively influenced other performances such as throughput and costs.

One of the main reasons for implementing QRM into the production process was to be more flexible in scheduling orders and as a result shorten the delivery time. According to Suri (2010) QRM is a strategy for reducing lead times throughout the complete organization. Cornerstone of QRM is the assertion that idle time (i.e. products spending time in buffers or inventory) is costly, in some cases even as costly as production time. More precisely, QRM claims that minimization of the Manufacturing Critical-path Time (MCT) will always result in cost savings.

Traditionally the company's focus was on machine utilization. That is, a production employee always operated the same machine. A single production order, which had to undergo multiple operations first entered the queue to undergo the first production step. The orders within the "queue" in front of the machine, operated by a permanent team, were handled one by one. This could lead to significantly large waiting times as the number of production orders rose. After a single production order underwent the first step it entered the "queue" in front of the second machine operated by another crew. In other words, between each production step a buffer existed.

However, within the time-focused approach all buffers between the different production steps were eliminated. A team of three employees executes all the production steps for a single production order consecutively. Next to process changes, introducing QRM also meant cross-training employees such that a single employee is capable to execute all steps involved to produce the end product.

2.2 Production Process

Once a request for production has been received by the production department the planner, located in Vianen, determines when and how (i.e. which machines are needed to acquire the product) the request will be fulfilled. Around 60% of the orders are completely handled in Vianen, while the remaining orders are partially or completely handled in The Hague. On average 1900 orders pass through the facility in The Hague each year, from which 70% are completely handled at this production location and 30% are transported to Vianen in order to be painted.

The production facility in Vianen handles a broad range of products, mainly on customer request whereas the production facility in The Hague mainly produces glazing beads, used to hold a pane of glass in its place. Although the number of variants are almost limitless, due to customized production, the production process of glazing beads and products related to those almost always has a fixed sequence of task executions. These are successively:

- gathering the raw materials needed and executing a quality check;
- shortening the gathered packages of wood;
- sawing the beam into multiple strips;
- planing and profiling the strips.

Before the actual production can take place, raw materials need to be gathered of sufficient quality to fulfill the production request. As wood is a product of nature the inventory of beams is of different quality and dimensions. Picking raw materials with the right quality is an essential part that should not be underestimated. In case the quality of the gathered materials is insufficient a greater loss of products will be encountered throughout the production process. Contrariwise, if the condition of the chosen beams exceeds the requested quality unnecessary costs are made, as quality of raw materials comes with a price. The process of gathering raw materials is not restricted by any means other than the use of a forklift. Furthermore, each team has a forklift at its disposal and therefore a team can always start gathering raw materials without waiting for equipment to become available. As the shop floor and the raw material storage are separated, transfer time is incurred when a team has to gather materials. This can be mentioned as one of the inefficiencies of the current work method.

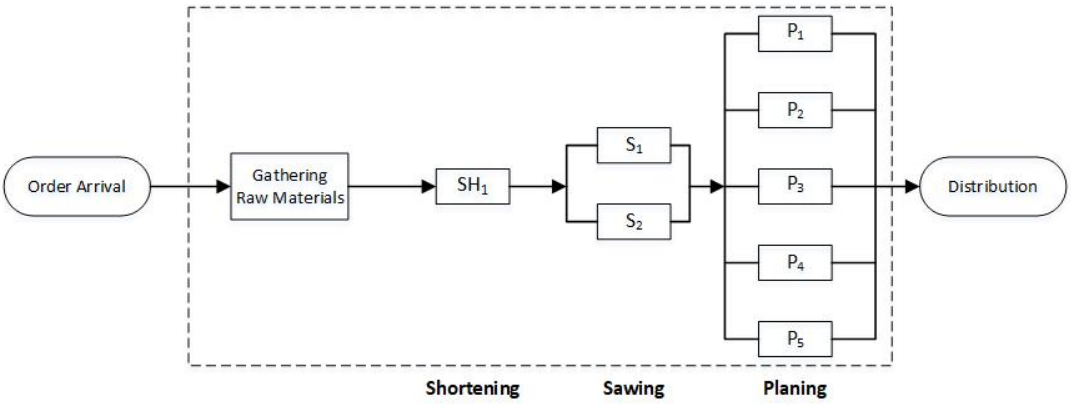

Figure 2.1: Flow model of the production facility in The Hague

After picking the needed materials the beams are shortened in case their length does not comply with customer's requirements. To do so, Royal Dekker has a single package saw at its disposal. Note that this step is omitted if the length of the beams already meet the requirements. Another production step that is not necessarily needed in case dimensions already match those required is sawing. This particular step is executed to adjust the thickness and/or width of a beam. Simply put, the beam is divided into multiple parts called strips. The process of sawing can be repeated as many times as necessary, depending on the number of strips needed in certain dimensions. As sawing is more time consuming than shortening two machines are available to perform this operation, assumed to be equal in terms of both capacity and speed. This is denoted in Figure 2.1 by splitting the flow into two individual paths.

Lastly, the strips are planed whereupon the product takes on its desired profile. Planing can be identified as the core process within the production department both because in this process the product receives its final shape and during this process the most value is added to the product. Over 98% of the products pass through a planer, whereas the remaining few percentages leave the factory as semi-finished products. A total number of 5 planers are available at the production plant in The Hague. These planers cannot be assumed to have the same requirements, some do have a higher maximum throughput rate. However, Royal Dekker delivers products of excellent quality and quality can be assumed to decrease in case the processing speed passes a certain rate. This rate is far behind the maximum throughput of the newly purchased machines. Since, quality needs to be excellent and operators should keep up with the process these two factors determine the throughput rate. It can be assumed that each planer operates at the same processing speed and set-up times are equal.

The order flow as described above is the most comprehensive path a production order can encounter. For some products it is possible to skip certain steps or to extend the path with a painting job in Vianen (for a detailed analysis see Appendix A). With a few exceptions, three employees take care of all the steps a single order has to undergo. In case of sawing and planing operations one operator feeds the machine and two others check the quality and tie up the products.

3 Research Design

This section describes the problem addressed in this research in more detail. Firstly, the problem encountered by Royal Dekker is described and possible bottlenecks are named. Thereafter the scientific relevance of the problem is discussed and some related papers are quoted. Based on both the problem description and the literature discussed research questions were formulated in section 3.3. Lastly, the problem is delimited and critical assumptions are elucidated.

3.1 Problem Description

Although QRM had significantly improved the throughput time of production orders it did not solve all problems encountered by Royal Dekker. These problems include low on-time delivery performance, high production costs per unit and in some cases a low throughput per unit of time. In the remainder of this report, on-time delivery percentage is referred to as the percentage of production orders finished before their original due date regardless whether or not the customer receives the products on-time due to poor logistics performances or an out-of-stock situation due to higher customer demand. In other terms, on-time delivery percentage is solely related to the performance of the production department.

As for the on-time delivery percentage of the production department the average goal is set to 85%, but the actual performance nearly reaches 70%[1]. This key performance indicator (KPI) exists of two individual measures, one related to the performance regarding customized orders (i.e. directly delivered to the customer) and one related to the performance regarding stock keeping units (SKUs). The goal and actual performance of the first are 95% and 73% respectively and for the latter 75% and 66% respectively. The goal of SKUs is implicitly lower than the aimed performance on customer demand, as some safety stock is in place and delivery to customers is not believed to be affected by short delays within the production process. Directly related to the poor on-time delivery performance are multiple complaints from customers, as well as lost earnings due to customers dropping out. Although this drop out of customers leads to some financial damage the largest costs are related to fines. These fines are incurred due to penalty clauses agreed upon with some customers in case of postponed or cancelled deliveries. At the end of 2018 a total amount of 385.000 € has been paid in fines. Although the underlying causes of these fines were not separated in detail, the majority is believed to be related to production specific problems.

Another issue mentioned by Royal Dekker are the high operating costs of the plant. According to the CEO the equipment at the shop floor should be well enough to satisfy demand on-time. Although, the package saw (denoted by SH_1 in Figure 2.1) can be indicated as a possible bottleneck, waiting time of teams (i.e idle operators) should easily be prevented by efficient scheduling methods. Although the production process is relatively simple, Royal Dekker puts a lot of effort in scheduling the jobs. A production planner spends time on managing the order flow, while the detailed planning is handled by two supervisors at the shop floor. One can easily imagine that the current planning process is time-consuming, as multiple layers are involved, leading to high (overhead) costs. Royal Dekker still fully support the ideas behind QRM, but realises that the current scheduling methodology does not support their needs.

The main drawback of the current production method, however, is related to the throughput per unit of time. While the cycle time has been reduced, the throughput rate has declined. To the best of the authors knowledge, this can be identified as the main cause of low on-time delivery performance. As a result of the low throughput rate, costs per unit rose since the equipment and number of employees remained unchanged after the transition to a QRM-based manufacturing process. In other terms, introducing QRM decreased the cycle time but resulted in a lower throughput per time unit and increased the production costs per product.

[1]Source: Production Report September 2019

3.2 Scientific Relevance

Although the problem faced by Royal Dekker includes multiple company specific aspects, by its very nature the issues related to scheduling can be identified as a job-shop scheduling problem (JSP). Formally, JSP is an optimization problem in which jobs are allocated to available production resources to satisfy some set of criteria, mostly aimed at minimizing the makespan. The problem is NP-hard and belongs to one of the most difficult combinatorial optimization problems (Lenstra and Rinnooy Kan, 1979). Many papers have studied the job shop problem, the majority of them using heuristics rather than solving the problem to optimality.

Most of the problems encountered in practice cannot be modeled as a (classical) job shop, due to additional aspects like limited storage space, (identical) parallel machines and multiple-resources. To take these restrictions into account a bunch of variations on the classical JSP have been researched such as the job-shop with sequence-dependent setup times, the hybrid job-shop, the open-shop, and the hybrid flow-shop. A flow-shop is characterized by an uni-directional routing of jobs through the shop, whereas in a job-shop the routing is not restricted by any means. Graphically shown in Figure 2.1 the production path at Royal Dekker satisfies the single direction flow of orders and can, therefore, be marked as a flow-shop. Furthermore, the problem can be named a hybrid flow-shop as some of the production steps can be executed on multiple (identical) machines. Since the problem not only faces the efficient scheduling of machines but also the deployment of teams it can be noted as a multi-resource hybrid flow-shop. That is to say, a single operation requires both the use of operators and a machine simultaneously.

Directly related to the current QRM-based production method, a single team executes all the production steps for a particular job consecutively. This entails that in case the next machine is not available, due to another job being processed, the team waits until the first machine capable of executing the operation becomes available. In literature these kinds of waiting times are referred to as blocking constraints and arise when buffer capacity is limited or even absent. Closely related to blocking is the no-wait restriction, where instead of blocking jobs are scheduled in such a manner waiting does not occur (i.e. delayed starting time). In a flow-shop with blocking, a job may occupy resources until the machine for its next operation becomes available. Considering the problem researched in this thesis, the team will be blocked but the previous machine will be available for production. We refer to this situation as partial blocking. Besides, both the makespan and total tardiness will be simultaneously optimized as both are key parameters in our problem.

Summarizing, this thesis deals with a multi-resource hybrid flow-shop with partial blocking constraints. Scientific papers addressing this special case of a job shop are scarce, whereas the development of heuristics to solve realistically sized problems is even more limited. To our beliefs, though, this problem occurs in various industrial areas. A heuristic solving the no-wait flexible job-shop in a glass factory is described by Alvarez-Valdes et al. (2005). To solve the problem of selecting machines for a workshop where aircraft components are produced, Mati et al. (2011) propose a genetic algorithm solving the flexible job-shop with blocking constraints. Production planning in a bakery is optimized by Hecker et al. (2013), who describe the problem as a hybrid no-wait flow-shop. Abyaneh and Zandieh (2012) proposed a genetic algorithm to solve the bi-objective hybrid flow-shop with sequence-dependent setup times and limited buffers.

However, to the best of our knowledge, no heuristics have been developed to optimize the bi-objective multi-resource hybrid flow-shop with partial blocking. Papers identifying some of the aspects of this problem most often test their heuristics on small problem instances, making them less valuable from a practical perspective as their computation time increases dramatically with the problem size. This thesis aims to narrow the gap between theory and practice by developing multiple heuristics, making a trade-off between solution quality and computational efficiency, to solve large problem instances of a highly realistic flow-shop while minimizing the number of assumptions.

3.3 Research Questions and Methodology

1. How to improve the current QRM-based production process keeping the necessary elements to guarantee short delivery times?
   (a) Can the size of the teams be changed?
   (b) What is the optimal team composition?
2. How should production orders be scheduled such that costs are minimized respecting both the fixed on-time delivery percentage and the QRM-based production process?
   (a) What is the optimal on-time delivery percentage?
3. Can performances be further improved when loosening some QRM-based restrictions creating a partial QRM-based process?
   (a) What elements of the production process have to be changed to create a partial QRM-based process?

3.4 Scope

To limit the research area of this report the problem is scoped. Firstly, the machinery is assumed to be satisfactory to meet all demand. In other terms, the average service rate will be seen as sufficient to process arriving jobs, whereas the scheduling (and thereby the set-up and cleaning times) of orders is denoted as the core problem. Increasing (or decreasing) the capacity of the machinery is left out of scope and is noted as a future research possibility. Furthermore, the number of employees is assumed to be fixed and cannot be identified as a decision parameter.

The determination of an alternative production path is not part of this research. Hence, the flow through the factory is known beforehand and cannot be changed. Sawing on a planer is only possible for a limited number of products. As this option is always faster than sawing on the saw, and the capacity of planers is far beyond the actual required capacity, it is always chosen as the best option. Although it is possible to switch to a normal saw, this is never done because of the above. Furthermore, each team has a forklift at its disposal, therefore it is assumed transferring between machines can be neglected as a forklift is directly available and a member can do this task during the setup time. This, however, does not hold for gathering raw materials, as the raw materials inventory and shop floor are physically separated.

4 Data Collection

Although heavily underestimated, data collection and analysis is a key component of operational research. It is through data collection that high quality, both in terms of reliability and validity, solutions can presented. In case of this research multiple data aspects need to be covered in order to create a high-quality and realistic scheduling method. Two main aspects of information are required to get a basic understanding of the production process, that are demand parameters and process parameters. The first can be best denoted by a demand distribution with a corresponding arrival rate. The second aspect yields parameters such as set-up times and throughput per unit of time. A third part of information required for this analysis, though not necessary needed for understanding the process, is a real life data set for testing different production methodologies.

4.1 Demand Parameters

One might mistakenly draw the conclusion the problems encountered by Royal Dekker may only be due to flaws of the production process whereas high variation in the arrival rate of orders might as well be a cause. As Cachon and Terwiesch (2013) state "Waiting time occurs when the expected demand rate exceeds the expected supply rate for some limited period of time. This leads to implied utilization levels of over 100 percent for some time period." Speaking of demand entails multiple parameters such as the arrival rate of orders and the magnitude of a specific order (i.e. service time). Both are important figures, the latter however is far more difficult to determine as the service time not only depends on the volume but also depends on the dimension of the end-product, the path through the production department and the output rate of the machinery (i.e. division into multiple strips). For now the service rate will be a black box and the order sizes of the real-life test data set will be assumed to depict the average order size and variation. In other terms as stated earlier the total capacity of the machinery is assumed to be satisfactory to meet demand.

The arrival rate of production orders was determined by analysing all orders handled by Royal Dekker in 2019. A total number of 255 days have been examined (excluding week 52 since the production department is closed during this period) in which 2055 orders have been produced. From this data it can be concluded that on average 8 production requests are received at the facility in The Hague, the minimum was 0 and the maximum number of orders on a single day is equal to 24. The data does not reflect the actual delivery date but the due date of the production request which actually is the delivery date minus 1 day. This for example entails that an order to be delivered on Monday has to be finished on the previous Friday. As the production department focuses on the due date only from now on we will refer to this date as if it where the delivery date.

To analyse the data correctly inaccuracies had to be filtered out. During the first analysis it was noticed that on average the demand on Fridays was remarkably higher than the number of orders encountered on other days. This could partially be explained by the given that orders to be delivered on Monday are due on Friday (as explained earlier). Some of the demand, however, was to be delivered on Tuesday but was given a due date on Friday by the sales department "in order to be sure the delivery date was met". These orders were identified and their due dates were corrected. This let to the arrival rate depicted by bars in Figure 4.1.

Although the data shows a wide spread of arrival rates, 90% of the rates lie between 1 and 15 orders on a single day. A mathematical distribution needs to be fitted such that a scheduling method can be created based on this demand distribution. Often the arrival process of customers/orders can be described by a Poisson process. A Poisson distribution with an expected value of 8, the sample mean, was fitted to the observed arrivals. Depicted by the smoothed curve in Figure 4.1 the Poisson distribution looks to fit the actual data quite well. A Chi-square (goodness of fit) test was performed on the data with 24 degrees of freedom. The χ^2 value is equal to 11.92, which is to a large extend due to the right tail of the observed values. The probability of having a value larger than 20 is less than 0.001 for a Poisson distribution with an expected value of 8, whereas

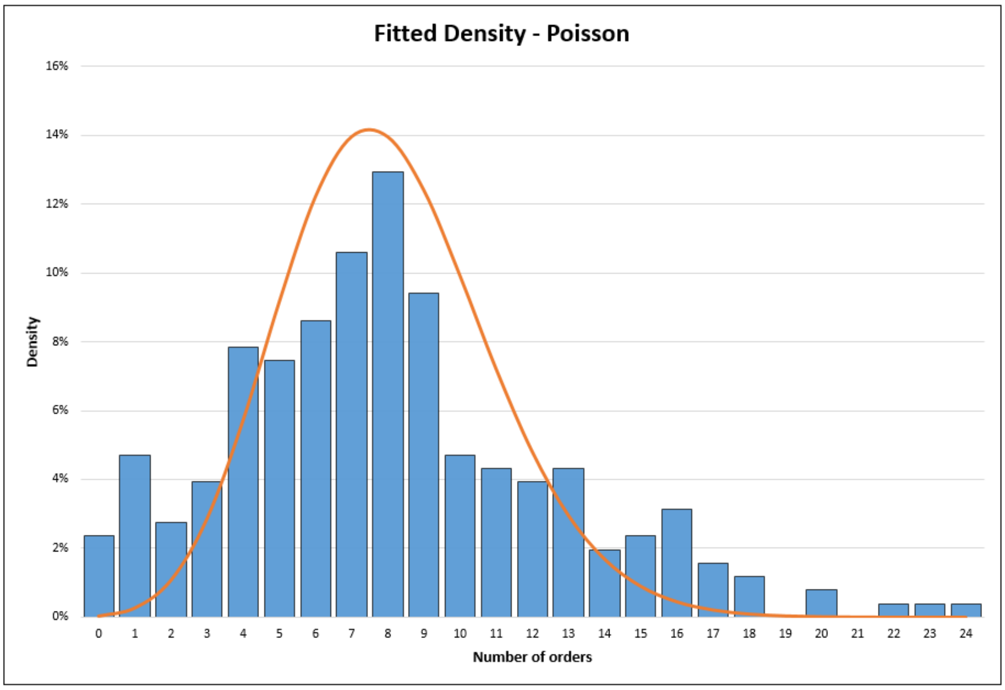

Figure 4.1: System overview

the probability of 20 or more orders arriving on the same day in the observed values is 0.02. This difference in the probability of a large number of arrivals accounts for 9.08 of the Chi-square value. To guarantee a valid result a p-value of at least 0.95 is necessary, which corresponds with a maximum Chi-square value of 12.40. The exact p-value equals 0.98 indicating the difference between the observed and expected values is considered to be not statistically significant and thus there is no reason to reject the Poisson distribution as being the arrival rate distribution.

4.2 Process Parameters

In order to fulfill the demand production takes places subjected to a number of parameters such as throughput of the machinery, set-up times and cleaning times. In some sense process parameters can be called supply parameters as those indicate the rate at which the demand will be fulfilled. As will be explained in more detail later orders need to be handled by at least 2 employees. Besides, handling orders with more than 3 employees would by no means decrease the production time. Hence, process parameters were determined for teams composed of 2 and 3 employees only.

In terms of throughput per time unit, teams of size 2 or 3 differ from each other on two subjects. Firstly, the average processing speed is decreased while working in teams of 2. Setup times and processing speeds for teams with 3 and 2 operators can be found in Appendix B, Table .2 and .3 respectively. These figures are made available by the commissioning company and can be marked as reliable, since they are based on thousands of orders. Processing speeds of both sawing and planing are on average reduced by 15%, whereas the time required for gathering and shortening stay untouched. The latter is directly related to the fact that the process of gathering and shortening the product can be taken care of by two workers. During this time the third employee usually cleans the machinery or starts the set up process for the next production step.

Secondly, a team composed of three is more flexible in terms of working at different processes simultaneously. As already mentioned some processes are handled by two employees rather than by three. During that time the third team member can execute tasks such as cleaning the machinery previously used or sets up the machine next in line. Moreover, after sawing or planing the machinery needs to be cleaned, related paperwork needs to be filled in and the finished products need to be wrapped. These tasks combined take on average 15 minutes. In case of teams composed of three employees these tasks and the set up of the next machine or gathering wood for the new order can be performed simultaneously. However, teams of 2 have to execute these steps consecutively.

To sum up, by downsizing teams from 3 to 2 employees time is saved since more teams can be composed, and therefore more orders can be handled at the same time. On the contrary, time is lost due to decreased machine throughput and the possibility to execute tasks simultaneously.

4.3 Test Data

To determine the effectiveness of different production methodologies these should be tested on a real data set. Testing methodologies on a single data set would not result in reliable outcome, this single data set though will be used as a base from which other test data can be created randomly.

Moreover, the orders required by Royal Dekker from the 3th of June till the 28th of June 2019 (20 weekdays) have been gathered such that the total data set entails 202 production requests. The complete list of orders including their volumes and operations can be found in Appendix C. The workload of these orders are slightly more than the current capacity (4 teams of 3 employees each) not considering the possibility to execute task consecutively (i.e. 15 minutes for each order). To be more precise the workload is equal to 101.1%. without the possibility to execute task consecutively. Subtracting the 15 minutes for 198 orders (first orders are excluded) a total of 49.5 hours will be saved resulting in a workload of 93.4%. Notice that waiting time, encountered when all equipment is used, are not considered calculating the workload.

The data set entails a varied mix of orders. The smallest order has a service time of 48 minutes, whereas the largest one keeps a team busy for over 10 hours (633 minutes). The average service time is equal to 193 minutes, however the corresponding distribution is right skewed indicating smaller orders are more common. This is also shown by the median being equal to 156 minutes. With a standard deviation of 135.7 minutes one can conclude service rates are highly varying.

Although the above gives some insight into the service times, these are heavily dependent on multiple parameters. The raw material dimensions not only determine the processing speed but also which equipment need to be used. For example, in case the requested length is not directly available an extra production step is added where the raw material is shortened. Besides, the production path is related to the type of wood being processed. Depending on the requested quality and dimensions a product can either be planed one by one or consecutively (up to 5 products simultaneously). In other terms there is not a homogeneous end product neither is it possible to determine a standard production path. As denoted earlier Appendix A shows 10 order flows and their steps. Appendix B indicates a single production step can have up to 5 different processing speeds depending on the product's dimensions. Combining both analyses over 50 different service rates can be distinguished. Adding the possibility a single production step has to be done twice, for example when the raw material dimensions are large a beam has to be sawn twice, more than 100 service rates are applicable to this production process.

Both, because the available data is not sufficient to make reasonable and well-founded assumptions about the service rate (and corresponding distribution) and because determining those rates would be a study on its own they are left out of scope. Therefore, the 202 orders represent all service rates on which analysis are build. However, to strengthen the quality of the results these orders were used to create 60 data sets. To do so, for each data set the delivery date for each single order was randomly determined on an interval from 1 to 20. All 60 data sets do therefore have the same workload, but differ in terms of due dates.

5 Improvement of the Production Process

In this section, an answer is provided to the first research question: *How to improve the current QRM-based production process keeping the necessary elements to guarantee short delivery times?*

The research question mentions that certain elements of the production process cannot or may not be changed. This restriction has been issued by the commissioning company as they experience those elements as positive and beneficial, although not scientifically proven. Among those elements two in particularly restrict the changes that can be made to the production process, that is to say, working in teams and multi employability. Working in teams has, according to the commissioning company, proven to lead to more flexibility and broader educated employees. Whereas multi employability has strengthened this flexibility and ensured each production request can be fulfilled without being dependent on the presence of certain machine operators. Moreover, a production request is handled by a team and this team executes all steps necessary to end up with a final product.

The introduction of team-based manufacturing at Royal Dekker is driven by fierce competition from companies abroad. By implementing team-based manufacturing they aim at reducing lead times, addressing the needs of customers to supply smaller quantities with delivery times getting more tight. These demands are common these days, while manufacturing enterprises are struggling with competition many of them see the need for designing flexible and agile systems that are able to deal with unpredictable demand patterns and a high variety of products (Abdul-Nour et al., 1999). Although the effectiveness of team strategies within manufacturing environments is disputable according to some researchers, the majority of papers support the introduction of self-managed teams. According to Elmuti (1997) these teams serve as the main building blocks of production organizations, increasing motivation, quality, productivity, and customer satisfaction.

Considering the restrictions issued by the commissioning company and the positive effects found in literature by multiple case studies, working in teams is seen to be an effective approach in coping with the highly varying demand and small production amounts. This limits the search space on how to improve the current QRM-based production process to determining the effective team sizes during day-to-day production. Therefore a sub question needs to be answered first; *Can the size of the production teams be changed?*. The remainder of this section is structured as follows: firstly, data of the current production process is analyzed with respect to different team sizes. Thereafter multiple production methodologies are described varying both the team size and the selection procedure of orders. The presented methodologies are then simulated by using procedures described in chapter 6 and a discussion on the results can be found in chapter 8.

5.1 Team Size

Production teams are currently composed of a fixed number of three employees. Based on the tasks a team needs to execute this team composition was chosen to be the most effective one. This decision was, however, at that time only based on physical workload and one did not look into the option of working with flexible team sizes. Current production scheduling aims at distributing the workload in such a way teams of three can be formed. Except for some non-influenceable situations such as illness of a team member all individual steps of the production process are executed by three operators.

Analyzing the tasks to be performed in greater detail, the minimum number of workers required (i.e. the lower bound on the team size) was determined to be two. All processes, except one, require at least two operators. Gathering raw materials involves lifting beams, of reasonable weight, making it hardly impossible for a person to solely execute this task. Shortening packages can be performed by one employee as it involves no manual labor as the process is highly automated. Processing time decreases, though, by working together as one can operate the package saw while the other employee operates the forklift. Sawing and planing are highly similar in terms of task execution. Both require a product specific set-up, a single employee to feed the machine and at least one operator checking the quality and stacking the products.

On the other hand, the upper bound on the team size is fixed to three. Firstly, it is important to mention that the gathering of raw materials has to be done in pairs, as a third member cannot support in checking the quality or at least will not speed up the process. Although a third team member can already do the set-up of the next machine, in case the machine is available, during the remaining processing time his capacity is not well-used. The same reasoning holds for shortening packages. As for, both sawing and planing the throughput rate is increased while working with three employees. Underlying reason is that quality checks, at the back of the machine, are performed twice as fast since the workload is distributed between two operators. Adding an extra employee to a team of three would by no means increase the throughput as the maximum speed is either determined by the boundaries of the equipment or the minimum quality required (i.e. from a certain speed level the quality can be assumed to decrease).

Based on the above, orders can be classified as either an order that can be handled by 2 operators or an order that needs 3 operators. Categorizing jobs is based on two factors, shown in Table 5.1. Weight is a major factor in whether one operator can sort the finished products after processing or two operators are needed to lift the weight. The wider the strip, the heavier the product and the more problems a single operator will encounter. Based on experiences of the operators themselves and their managers an assumption was made that end products being less than 90mm in width can be handled by one employee whereas wider products are assumed to be too heavy such that two operators are needed. Secondly, as explained before, during the process a beam can be split into multiple pieces, once the input is split into three or more parts a single operator cannot keep up with the (already reduced) speed of the machinery. Both imply the need for a team of three rather than a 2-operator team.

Table 5.1: Classification of jobs based on team size

Operators	**Weight**	**# Sorting**
2	Width $< 90mm$	1 or 2
3	Width $\geq 90mm$	≥ 3

Following the classification rules depicted in Table 5.1 jobs within the data set were indicated as either a 2- or 3-operator job. That led to a total number of 94 jobs (47%) restricted to a team size of three employees and 108 jobs (53%) to be processed by at least 2 operators. By far the most frequent restriction for classifying jobs as 3-operator orders was the number of simultaneous output as shown in Figure 5.1. In four cases both the product had a width greater than 90mm and the simultaneous output was above 2 pieces.

Effects of working in teams of 2 are analyzed by doing detailed calculations on an average job (Table 5.2), that is to say, the related production path is the most common one and the required output meters are around average. Firstly, in total 3.6 m^3 of raw materials have to be gathered. The next operation to be scheduled is sawing (task nr. DH-ZA-K010) subjected to an input of 1113 meters. During this operation beams are cut once, leading to an input of 2226 meters for the next task, profiling 2 parts simultaneously. Total processing time, excluding time required for cleaning, for this particular job is 363 and 323 minutes for teams with 2 and 3 operators respectively.

Table 5.2: Comparison of team size throughput, with speed in *units/min* and setup time in *min*

	Operation 1 - Gathering			***Operation 2 - Sawing***			***Operation 3 - Planing***		
Team Size	**Speed**	**Setup**	**Tot. time**	**Speed**	**Setup**	**Tot. time**	**Speed**	**Setup**	**Tot. time**
2 operators	0.07	0	52	12	16	109	12	16	202
3 operators	0.07	0	52	14	16	96	14	16	175

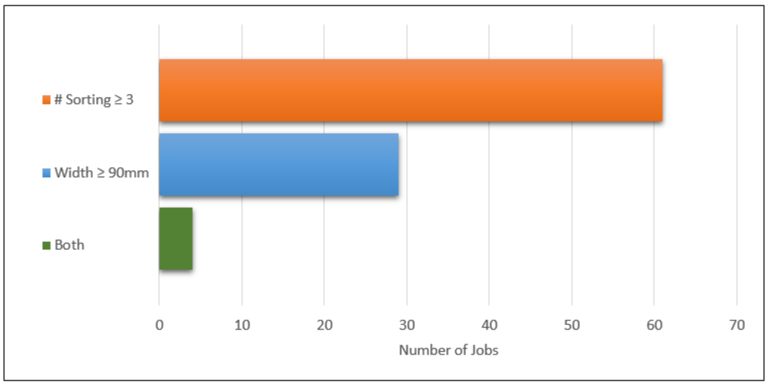

Figure 5.1: Number of orders to be processed by 3 operators by their cause

These processing times, however, say little about the actual man hours spend on a single job as the faster processing time of 3-operator teams is amplified by the simultaneous execution of setup and cleaning tasks. This also involves adding 15 minutes to the processing time of both operation 2 and 3. Timelines related to processing the job either by a team composed of 2 or 3 operators are visualized in Figure 5.2, where k_1 denotes the first processing stage (not mentioning the actual equipment used) and k_2 the second processing stage respectively. It should be explicitly mentioned these timelines only hold in case all required machines are available at the start of the operation. Total makespan of the upper schedule is 393 minutes, whereas increasing the number of team members reduces the completion time to 323 minutes.

During the job execution, the members of a small team will not perform tasks separately. It, therefore, can be stated that the job used in this example requires 13.1 hours ($2 * 393$ minutes) of labor. Concerning teams composed of 3 operators, they can perform tasks simultaneously, however, this is subjected to the availability of the next machine. In the most ideal situation, the first 15 minutes of raw material collection are done by 2 operators, whereas the third member cleans the machine previously used. A similar reasoning holds for the last cleaning task to be performed by one operator whereas the others start gathering the next job. Combining this information, the job is handled for 15 minutes by 2 operators, processed by three operators for 293 minutes, and the last cleaning step of 15 minutes is handled by a single employee. A total of 15.4 man hours are required to finish the example job. Note that the latter option requires almost 15% more labor.

Concluding, working in teams of three reduces the lead time of a single order which was a major reason for implementing QRM. However, it is also shown that by doing so more labor is required and inextricably costs per unit will rise.

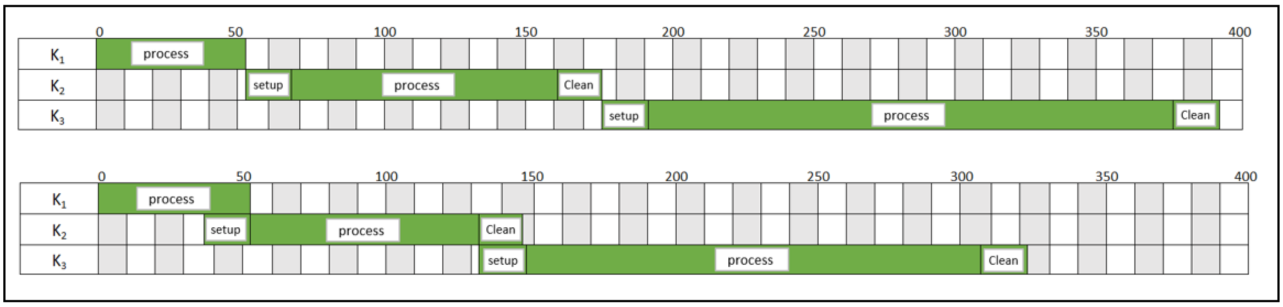

Figure 5.2: Gantt charts for both processing by a 2- and 3-operator team

5.2 Production Methodologies

A total number of 12 full-time employees work at the facility in the Hague. In order to make an equitable comparison between different team compositions, the total number of employees should not be varied. Although this limits the solutions space three team settings are suggested based on this occupation. These settings not only differ in terms of team size, either 2 or 3 operators, but also on the restrictions which order is picked next. The list of production requests is sorted based on 2 criteria. The first, and most important one, is the due date of an order. Second criterion is the order number, which is ascending. The latter is directly related to the arrival time of a production request. In other terms, the order list is sorted firstly on due date and orders sharing the same due date are sorted on their arrival time (FIFO). This to a great extent reflects the current manner in which production orders are scheduled. In the remainder of this section, all three settings will be explained in more detail both concerning the distribution of operators and the selection procedure of orders.

Fixed-3
The fixed team size of three employees with a total number of four teams depicts the current situation at Royal Dekker. A benefit of this team composition is the small probability of teams waiting for equipment to become available as it is been used by other teams. Compared to the other team settings this methodology has the lowest number of teams and the overall service rates are larger (i.e. teams of three do have higher processing speeds). Contrariwise, fewer orders are handled simultaneously, as there are only 4 teams, increasing the possibility of orders being delayed, especially the last orders processed for a single due date.

Whenever a team finishes their order the next order on the list is picked without any other constraint. As explained earlier the list of orders is sorted based on two criteria and in case of the Fixed-3 methodology simply the next order is selected as the distinction between orders that can be handled by either 2 or 3 operators does not impact the analysis.

Taking a closer look at the basis data set of 202 orders, the minimum required production time was calculated. Scheduling the orders according to the current sequencing method used, a makespan of 154.2 hours is achieved. For this example it suffices to know that orders are firstly assigned based on their due date (EDD-rule) and secondly on their arrival date (i.e. order number), for a detailed description we refer to section 7.1. Results obtained from this particular order instance show a total (combined) waiting time of 3.4 hours. Assuming the waiting times are incurred proportionally over all four teams, less than 1 hour of the makespan is due to nonavailability of machines. By slightly changing the order sequence it can be demonstrated that the waiting time can almost be reduced to zero. Hence, the minimum required production time can be calculated without taking waiting time into account, pointing to the teams (i.e. labor) as the actual bottleneck.

Summing the production time of all individual orders a total required completion time of 149.4 hours was observed, taking the simultaneous execution of setup and cleaning into account. This can be said to be the absolute minimum makespan of the Fixed-3 approach. Note that in practice a higher makespan is observed due to two reasons: 1) waiting time due to nonavailability of machines and 2) inefficient use of labor at the tail of the simulation. The latter refers to teams being idle as they have already finished, but still have to wait for the last team to complete its job.

Fixed-2/3
The Fixed-2/3 methodology represents a mixed setting with two teams composed of 3 employees and three teams composed of 2 employees. Main benefit of this methodology is directly related to the number of orders handled simultaneously. It is expected that the more orders are processed simultaneously, although at a slightly lower service rate, fewer orders will be delayed. This expectation is directly related to queuing theory in which a single server with service time μ is less effective than two servers both having a service time of $\frac{1}{2}\mu$.

Contrary to the Fixed-3 methodology selecting the next order from the list of outstanding production requests is subjected to an extra constraint. As mentioned in subsection 5.1 orders are marked as either handleable by 2 or 3 operators, in case a team of size 2 is finished the next order selected is the first on the list that can be processed by 2 operators. Teams composed of 3 employees follow the same procedure, except that they only pick orders selected to be handled by three workers. In either case, the orders for teams of 2 or teams of 3 have all been finished the teams concerned stay idle (or execute some side jobs such as cleaning the warehouse) until the other teams have finished.

The above way of selecting orders both has benefits as well as downsides. Main advantage is the effective use of operators since on no occasion more operators will be deployed as strictly necessary. The downside of such a fixed method is its high dependence on the ratio of orders handled by 2 or 3 operators. In case the percentage of orders requiring 2 employees is large the teams cannot keep up with the arrival rate and eventually end up with large delays. This likewise holds for the case a large number of orders arrive only handleable by teams of size 3. Next to being very sensitive to this ratio, teams staying idle at the end of the order list will affect the performance negatively. This effect, however, will become negligible in the long term as more and more orders arrive.

Again, a single test simulation was performed using the current scheduling rules. This time a distinction was made between the time required for 2- and 3-operator teams, with a workload of 407.5 and 264.3 hours respectively. Dividing the workload by the actual number of teams makespan for teams composed of 2 operators is equal to 135.8 hours and 132.1 for teams of size 3. Therefore, the absolute minimum makespan of the Fixed-2/3 approach is 135.8 hours since the latest finishing time counts. Compared to the Fixed-3 methodology almost 14 hours can be saved in the best case scenario. Extensive testing has to be done in order to observe the effect on tardiness.

Variable-2/3
The Variable-2/3 production method is with regard to almost all aspects similar to the Fixed-2/3 methodology. As for the mixed team setting all parameters are equal; two teams composed of 3 employees and three teams composed of 2 employees.

Different from the Fixed-2/3 methodology is the selection of outstanding orders by teams of size 3. Teams composed of 2 employees will still process the orders handleable by 2 operators one by one. On the contrary, teams with 3 employees pick the next order on the list whether it is suitable for teams of 2 or 3. In other terms, teams of size three support teams of 2 were needed. Whenever such an event occurs the order will be processed with service rates suitable to teams with 3 operators. Notice that compared to the Fixed-2/3 approach the outcome will be affected largely if the percentage of 2-operators only orders is high and stays (almost) the same in case the percentage of 3-operators only orders is high.

Compared to the Fixed-2/3 approach this methodology is less sensitive to the ratio of orders handled by either 2 or 3 operators. This method, however, is more vulnerable to the arrival sequence of orders. For example, in the circumstance that a large percentage of orders with an early due date are suitable for teams of size 2, the 3-operator teams will support by processing those orders. Once the percentage of orders only handleable by 3-operator teams with a later due date then turns out to be large these teams stay behind as they cannot keep up with the demand which could have been prevented by directly processing those orders.

Determining the minimum makespan for the Variable-2/3 approach is more difficult. This is due to the already explained vulnerability to the sequence of orders. Both clusters of teams do not have a fixed number of orders assigned to work on. In other terms, the workload is not fixed and depends on the arrival of orders. In the best case scenario, however, the somewhat longer completion time of 2-operator teams (i.e. 135.8 hours) can be reduced by assigning some of the workload to 3-operator teams that have by themselves a completion time of 132.1 hours. By balancing the workload an absolute minimum makespan of 133.8 hours was determined.

5.3 Conclusion

The main goal of this chapter was to provide methodologies that are aimed at reducing two criteria, makespan and total tardiness, while reducing the overall production costs. These methodologies are used as a building block to find a (sub)-optimal solution to the scheduling problem faced by Royal Dekker. It was noted that taking into account the argued restrictions and the positive effects found in literature regarding team-based manufacturing, the problem could be limited to finding an optimal size for the production teams. By analyzing the process it could be concluded the current team size of 3 operators could be reduced to 2 operators with restrictions to weight and sorting limits.

To partially answer the first research question (*How to improve the current QRM-based production process keeping the necessary elements to guarantee short delivery times?*) a small case-study was performed. This entailed the actual scheduling of a job from the data set provided by the commissioning company. Analysis showed labor can be reduced by almost 15% when an average job, satisfying the weight and sorting constraints, is handled by two employees rather than by three. Based on this finding two alternative production methods, Fixed-2/3 and Variable-2/3, were proposed. It can be stated that overall costs decrease and thus improving the current production process in that aspect. As both makespan and total tardiness values also depend on the waiting times incurred a simulation is required to answer question 1b (*What is the optimal team composition?*). The results of these analyses are presented in chapter 8.

6 Model Description

The process information mentioned in the previous chapters has been translated into a mathematical model. As it has already been mentioned the problem under investigation can be labeled as a bi-objective multi-resource hybrid flow-shop with partial blocking the features of it are only described very briefly. Whereas the main focus of this section is on presenting a mathematical formulation and describing the simulation procedures. The remainder of this chapter is as follows, firstly a mathematical formulation of the problem is provided and important modelling assumptions are motivated. Afterwards, a timetabling procedure for assigning resources to operations in a QRM production environment is proposed. Section 6.3 describes a timetabling procedure for a modified flow shop by introducing a limited buffer. Lastly, the model is validated and verified.

6.1 Flow-shop Model

In a hybrid flow-shop a set of n jobs, $J = \{j_1, j_2, ..., j_n\}$, which have to process through at least one and at maximum k stages. Each job consists of a maximum number of y operations, $O = \{o_1, o_2, ..., o_y\}$, owning a predetermined processing order through the manufacturing facility, that is to say the routing through the stages is fixed. Being an extension of the standard job-shop scheduling problem, the problem under consideration is clearly NP-hard. The following notations are used in this research:

n	the number of jobs to be scheduled ($j = 1, 2, ..., n$)
k	the number of serial stages
q_k	the number of parallel machines at stage i
$p_{i,j}$	processing time for job j at stage i ($i = 1, 2, ..., k$)
$C_{i,j}$	completion time of job j at stage i
C_i	completion time of job j
d_j	due date of job j
C_{max}	maximum completion time of a schedule ($C_{max} = max\{C_1, ..., C_n\}$)
T_j	tardiness of job j ($T_j = max\{C_j - d_j)$
π	permutation of the given jobs ($\pi = \{\pi_1, \pi_2, ..., \pi_n\}$)

6.1.1 Multiple Resources

In an extended version of the general shop scheduling problem, an operation may need several resources to be executed. This particular problem has been first addressed by Dauzere-Peres et al. (1998). Multi-resource problems are hard to tackle as it involves two extensions that increase the solution space dramatically. Firstly, *multi-resource* entails several resources to be assigned at the same time to an operation. One can easily imagine, a growing number of various resources to be used simultaneously lead to extensive calculations. Secondly, *resource flexibility* is needed as a resource will be selected from a given set.

As new decision variables are added to the scheduling problem, both an assignment and a sequence of operations on resources have to be determined. In case of a flow-shop with no buffer capacity these problems can be solved separately. This is a great advantage compared to a job shop, where deadlocks may arise and the assignment of jobs to resources directly influences the feasibility of solutions presented by the sequencing algorithm. In an unidirectional flow shop all sequences are feasible, although not all of them may be acceptable in terms of performance criteria.

As for the flow shop at Royal Dekker, the first step (i.e. gathering raw materials) only requires a production team, whereas for all other operations it is mandatory to both reserve a single machine at the corresponding stage and a production team. More specifically, the following notation will be used. A set of $R = \{R_1, R_2, ..., R_m\}$ resources exists, where m is the number of resources. For our problem instance this number equals the sum of all the stages k and the number of different team sizes either 1 (i.e. only teams composed of 3) or 2 (i.e. a mixture of 2- and 3-operator teams). Each resource is available in a given number Q_r, for the stages this number is equal to variable q_k. Resources required for each operation i of job j, denoted as $o_{j,i}$, are to be selected from a given set of candidates $F_{j,i} = \{A^1_{j,i}, A^2_{j,i}, ..., A^{m_{j,i}}_{j,i}\}$. The variable mj, i depicts the number of candidate resource sets of $o_{j,i}$. Within the assignment problem resources, in other terms candidate sets, are allocated to operations. As long as capacity restrictions are respected, feasible solutions are guaranteed.

6.1.2 Parallel Machines

As mentioned earlier the hybrid flow shop scheduling problem involves sequencing jobs in a shop with at least one stage containing more than one machine. The hybrid, or alternatively flexible, flow shop is a generalization of two well-known machine scheduling problems: the classical single-machine flow shop problem and the parallel machine scheduling problem (Wang, 2005).

The number of identical (parallel) machines in a single stage is denoted by q and all machines of a single stage are assumed to process at the same production rate. Processing time time for job j in stage i is described by $p_{i,j}$. Since machines are considered to be identical in terms of both setup times and processing speeds the complexity of the hybrid flow shop is reduced. Besides, the absence of buffers further reduces the problem complexity. This can be related to a limited number of possibilities of assigning an operation to a machine. Let us explain this by use of an example.

In case of unlimited buffers, the number of jobs at the shop floor can rise quickly. Let us assume in front of a certain stage a buffer exists of 10 jobs waiting to be processed. When an enumeration procedure is used the future paths of all of those orders have to be calculated in case job 1 is to be selected. Similar to selecting job 2, all future paths have to be recalculated, and so on. Under the assumption of no intermediate buffers the Work-In-Progress (WIP) is kept low and the maximum number of jobs waiting to be performed on a particular machine is equal to the number of teams minus 1 (i.e. the job currently being processed). Less jobs also imply less enumerations, thus, reducing the problem complexity.

6.1.3 Reentrant Jobs

Our problem is also subjected to some kind of recirculation, as jobs can re-enter at some stages. Re-entering is only possible at the sawing an planing stage in order to change the dimensions of the product and has to be executed consecutively (i.e. a job can not return to a previous stage). In this respect, the recirculation differs from the most commonly used definition. As defined by Ahonen and De Alvarenga (2017), in the context of flexible flow shops a job needs not to return to the same machine but to one of the machines of a previous stage. In our case the job cannot return to a previous stage. The reentrant operation, however, is not restricted to be executed on the same machine. It can well be processed on a different machine of the same stage. Note that the latter even is beneficial in terms of simultaneous setup.

Due to the fact a reentrant job can be scheduled on all machines of the corresponding stage, it can be handled as if it where a new operation. It is important to have a thorough understanding of these jobs. One could mistakenly suppose that two consecutive operations to be performed at the same stage are handled as a combined operation (i.e. adding up the processing times) and, therefore, is only scheduled once. That deviates from the actuality, because in reality a full setup has to be performed in between those operations. A job currently waiting to be processed on the same machine, having a larger priority, will be scheduled first.

6.1.4 Partial Blocking

The partial blocking constraint implies that a job, having completed processing at a certain stage, requires the team to wait until the next machine downstream becomes available. Although the job does not remain on the machine, and therefore the machine can be used for processing other operations, the blocking of teams is indirectly related to the limited buffer capacity at the plant.

A resource is either released after the completion of an operation or it is blocked to ensure it's availability. In our problem blocking is only related to teams. Whether a team is available or not is denoted by the binary variable Y_t where t denotes the team identification number. In case the binary variable is 1, a team is either being operational or it is blocked. At the moment a team is released, only in case all operations of a job are finished, Y_t is set to 0.

6.1.5 Assumptions

An important aspect of research is the level of consistency, this eventually results in a reliable conclusion. In case this study will be redone the same results should be obtained. Therefore all design decisions have to be mentioned. Main assumptions made by this research are:

1. All resources and all jobs are available from time $t = 0$.
2. Operators work 8 hours per day for 5 days a week.
3. Machines do not break down during operation and manpower of uniform ability is always available (i.e. no absence of employees).
4. No machine may process more than one operation at a time.
5. Next to the setup times both the sawing and planing machines need to be cleaned and administrative tasks need to be done after production. A team needs on average 15 minutes to do these tasks after production.
6. Teams of size three can split up during the set-up and cleaning, i.e. production tasks of these teams can overlap during set-up of the next machine or gathering materials for the next production order.
7. Teams of size two cannot split up during set-up and cleaning, i.e. each step needs to be executed consecutively.
8. Once an order has been started on a machine it cannot be interrupted by another order (non-preemptive scheduling).
9. The production path (job routing though subsequent stages) is given and no alternative routings are permitted.
10. Gathering raw materials takes at least 30 minutes. Once the calculated time (according to Appendix B) for this task is less than 30 minutes the service time is increased to 30 minutes.

With regards to the last point mentioned, this assumption is directly related to the physical distance between the raw materials inventory and the shop floor. A single team member drives the forklift towards the space where the required materials are located, whereas the other team members have to walk. Slightly dependent on the location of the raw materials on average 5 minutes are required to walk from the plant to the gathering area. Note that this "transfer time" is incurred twice, as the team has to walk back to the plant after the materials are collected. For larger orders, i.e. of high volume, this transfer time can be neglected as the processing times increase rapidly. However, the processing time does not reflect the actual time required in case of low volume orders. On average gathering 1 m^3 takes 15 minutes, but this number does not include the transfer time. Based on the actual experience of the shop floor supervisors it can be assumed gathering raw materials takes at least 30 minutes.

6.2 QRM-Based Timetabling Procedure

By definition the multi-resource hybrid flow shop can be said to be two-fold. More specific, the problem can be defined as two separate, though not independent, sub-problems. The first is related to the sequence of jobs, while the second problem is to determine the actual timetabling of operations respecting both resource and buffer constraints. In other terms, the timetabling method depicts the actual execution of the sequence determined by the sequencing heuristic.

Consequently, the assignment of operations to resources is a fixed procedure aiming to approximate the real execution of jobs throughout the plant as good as possible. The sequencing method can be a simple rule as well as an advanced search procedure. For now the sequence of orders, presented by permutation array π, is supposed to be known. This piece of information is one of the most important input values to the timetabling procedure as it starts scheduling orders from the top of the list downwards until all jobs are processed.

Algorithm 1 denotes the pseudo-code of the timetabling procedure. First of all, a future event set $\mathcal{F}$ is created in which all outstanding events are incorporated. Two arrays, $teamSize()$ and $teamStatus()$, are initialized at the start of the procedure. The first array includes for all teams, denoted by the variable $nrTeams$, the size of the particular team. Besides, $teamStatus()$ stores the activity of each team, which can take either the value *busy*, *waiting* or *idle*. In case a team is *busy*, the operators are actually processing an operation, which requires a machine too. *Waiting* refers to the situation a team is assigned to a job, but the next operation downstream cannot be executed because no equipment is available. Note that this state depicts the partial blocking constraint, where a team cannot process any other operation whereas the previous machine can still be used to process another job. At the end of the simulation some teams finish early, whereas others are still execution the last operations. Teams finishing and finding no jobs left to process become *idle*. Besides the number of q machines for each stage is defined, except for the gathering of raw materials as this requires a team only.

After the initialization of all parameters the first i jobs are scheduled, where i depends on the number of teams available. End time of the event can simply be calculated by adding the processing time $p_{i,j}$ of job j at stage i. Note that for the first operation this always equals the team for gathering raw materials as this step cannot be skipped.

Taking from $\mathcal{F}$ the first event, that is the event with the earliest time it is first checked whether the corresponding team is not waiting. If this condition is true, the team was busy and, hence the event denotes the end of an operation. Equipment that was used to execute the operation is released (i.e. becomes available). Teams, however, stay assigned to their job and are not yet released at this point. This is due to the partial blocking constraint imposed to ensure a team finishes all the operations of a single job consecutively.

Not depending on whether a team is either *busy* or *waiting* it is checked whether job j has operations pending. In turn, if the team was waiting the event taken from $\mathcal{F}$ depicted the end of the waiting time. Let us first consider the case not all operations of job j have been finished. Referring to reentrant jobs, the procedure does not differentiate between jobs passing on to the next stage or the ones re-entering the same stage again. If at least one of the machines in the stage is available job j is assigned to a particular resource and the end of the operation is calculated. Whenever all parallel machines are being used the earliest possible starting time is determined. This entails both calculating the remaining processing time of the job currently being processed and the processing times of all the jobs waiting to be processed on that stage. In other terms, job j is added to the back of the queue and has to wait for the first machine to become available.

In case all operations of job j are executed the job is removed from the sequencing list π. In the event that there are still jobs to be processed, the first job on the sequence list satisfying the team size restrictions is picked and a team starts gathering the raw materials. Whenever the team finds the order list to be empty the team remains idle. Termination is stopped once all teams reach the state *idle*.

Although, not specifically mentioned determining the end time of an event involves more than just adding the processing time to the current time. It also involves scheduling simultaneous tasks such as cleaning and setup. To save computation time a simple rule was applied: at the end of an operation it is determined whether the next machine was already available x minutes ago, where x is the minimum of the times required for setup and cleaning. If this condition holds the start of the setup is scheduled "in the past" and $p_{i,j}$ equals only the actual processing and cleaning time. Needless to say, this $p_{i,j}$ is only used for calculating $e.time$ and the actual registered (overlapping) processing times includes the setup time. If the machine was not available x minutes ago, simultaneous setup is not possible.

6.3 Two-Stage Timetabling Procedure

Royal Dekker fully supports the ideas behind QRM and attach great importance to working in teams. Important to them is the flexibility, in other words the multi-operability of operators in order to be less dependent in case of illness or vacation of certain employees. In chapter 5 it is already shown that great improvements are observed by adding more flexibility to the team size, still respecting the QRM-based production methodology. This process, though, is still subjected to some fundamental inefficiencies. Aim of this chapter is to compare the current QRM-based process with a flexible process, still taking into account a mixture of team sizes but loosening the constraint that all operations of a single order have to be performed by a single team. To support the claims made in this chapter, analysis of the model is performed in chapter 8 by applying the EDD-SPT algorithm that will be introduced in chapter 7.

6.3.1 Inefficiencies of the QRM-based Production Process

As mentioned above the current production process is subjected to multiple inefficiencies. In this section the problems are identified and explained. Afterwards we present a modified production methodology in section 6.3.2, which for a large extend overcomes the described inefficiencies.

First inefficiency is related to the physically separation of raw material inventory and the shop floor, as explained in section 6.1.5. Total transfer time equals 10 minutes, which could have been used to actually process products. By aiming to minimize the times a team has to travel from the plant to the storage facility a significant amount of production time could be gained. In case all of the 202 orders were gathered by a dedicated team rather than letting each team execute this task separately almost 34 hours are saved. Note that this are team-hours and one should not mistakenly assume the makespan will be reduced by the same amount of time.

Secondly, collecting raw materials based on the right quality is a two person task. That is to say, two employees are required to lift the beams and check their quality, whereas a third team member cannot or only to a very limited extend provide support. During the first and last 15 minutes of the gathering process a third team member is responsible for cleaning the machine used and setting-up the next machine respectively. When the actual operation exceeds 30 minutes only two employees are effectively working while the third is executing some side jobs. A quick analysis of the data sets learns almost 1/3 of the orders take more than 30 minutes to collect. More precisely, on this particular data set 60.3 man hours could be saved. Hence, a team composed of 3 operators will able to spend 20 hours processing products if these hours were saved. A possibility to overcome this burden is to always let a team of 2 employees gather the required materials.

Within each production process a bottleneck machine can be identified. Depending on the characteristics of the environment this is either the machine with the lowest processing speed (i.e. determining the speed of the entire process) or the piece of equipment with the highest utilization rate. For the problem under investigation both the shortening machine and both saws can be identified as bottlenecks. The first can be said to be a bottleneck since 70% of the waiting time is incurred during this stage. However, the waiting does not occur due to a high utilization rate, since the processing time is short, but rather because only a single machine is available and the probability of two teams requiring this machine at the same time period is reasonable. Again, by making use of a single team dedicated to collect the required materials waiting time will not be incurred anymore since only one team makes use of the shorting machine.

Algorithm 1 QRM-Based Timetabling Procedure

```
Create F = {}, teamSize(), teamStatus()
Set nrShort = 1, nrSaw = 2, nrPlan = 5, nrTeams = 4/5, time = 0
Set teamSize(i) = 3 or teamSize(i) = 2 for all teams

for i = 1 to nrTeams do
  assign π_i to team i and set teamStatus(i) = busy
  set end time of the event e.time = time + p_{i,j}
  add e.time to F
end for

while number of teams being idle < nrTeams do
  take first event e from F
  get time of event e and set time = e.time
  if teamStatus(i) <> waiting then
    register end time of the operation
    register the release of the equipment, depending on the resource used (nrShort =
    nrShort + 1 OR nrSaw = nrSaw + 1 OR nrPlan = nrPlan + 1)
  end if

  if job j has pending operations then
    if machine is available then
      reserve the machine (nrShort = nrShort − 1 OR nrSaw = nrSaw − 1 OR nrPlan =
      nrPlan − 1)
      set end time of the event e.time = time + p_{i,j}
      add e.time to F
      teamStatus(i) = busy
    else
      determine the earliest time a machine becomes available
      set end time of the waiting event e.time = time + w_{i,j}
      add e.time to F
      teamStatus(i) = waiting
    end if
  else if job j has no pending operations then
    register end time of j and calculate its delay
    remove j from π
    if π <> empty then
      pick the next job of π that satisfies the team size restrictions (i.e. teamSize(i) is equal
      to the number of operators required)
      set end time of the event e.time = time + p_{i,j}
      add e.time to F
      teamStatus(i) = busy
    else
      teamStatus(i) = idle
    end if
  end if
end while
```

6.3.2 The Two-Stage Production Process

To (partially) remove the inefficiencies and thereby increasing the effective production time a two-stage production process is proposed, while still respecting working with teams. As can be seen in Figure 6.1 a buffer has been placed in between the shortening of raw materials and sawing of beams. As for the flow shop at Royal Dekker, the location of this buffer is relatively straightforward as valuable time is saved by doing so. The allocation of buffers, however, is of great complexity. Main trade-off coped within such a problem is between increasing throughput and reducing work-in-process time.

Vouros and Papadopoulos (1998) investigated the optimization of production lines by determining a near optimal buffer allocation plan. Their objective was to maximize throughput of the production line. Interestingly, it was concluded that the buffer capacity affects shop performance but the performance improvement diminishes rapidly with increased buffer size. Although the aim of this research is not to find the optimal buffer size, the problem will be extended to multiple configurations that differ in terms of buffer size.

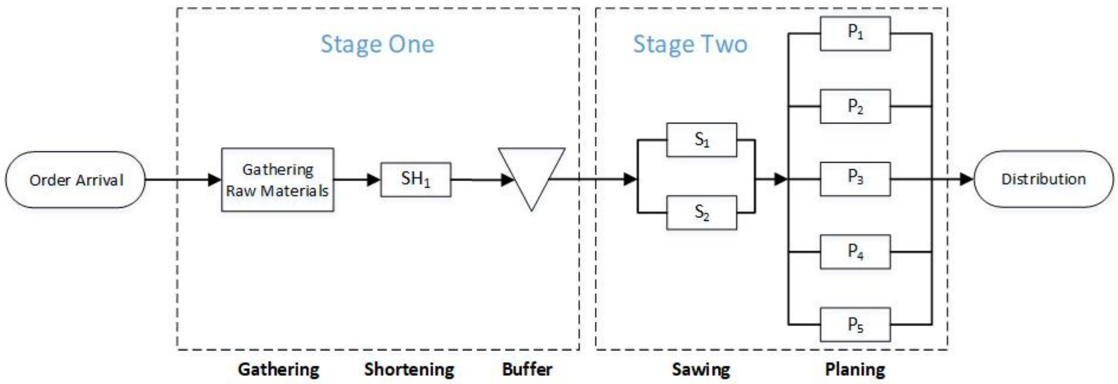

Figure 6.1: Flow model of the two-stage production process

Let us first explain the two-stage production process and more importantly the differences with the QRM-based method. Algorithm 2 shows a pseudo-code of the timetabling algorithm for our two-stage flow model. The procedure in a broad sense can be seen as an extension of the QRM-based timetabling algorithm. Whereas, the latter distinguishes production teams only, the two-stage approach starts with a single team dedicated to performing operations at the first stage, from now on referred to as buffer team, and stage-two teams mainly perform sawing and planing operations. At first, the buffer team starts performing all tasks of stage one (i.e. gathering and shortening raw materials). In either case, the maximum capacity of the buffer is reached or all orders have passed through stage one, the buffer team starts processing orders in stage two.

A great advantage of the two-stage process is the (partial) elimination of transfer times. Although a single member of the buffer team still has to bring the gathered materials to the buffer location by using a forklift, the time required to do so is already included in the processing time (15 minutes per $1m^3$). Actual transfer times were incurred due to the other team members walking to the raw materials inventory, which is not required anymore. This absence of physically moving towards the inventory saves in total 10 minutes on every individual order. Hence, in case jobs are taken care of by the buffer team 10 minutes are subtracted from the actual processing time. Note that, the minimum required time for gathering raw materials is thereby reduced to 20 minutes.

Depending on the maximum capacity of the buffer, the dedicated team processes a certain amount of orders at stage two. Similarly, normal production teams perform a certain percentage of operations in stage one depending on the buffer capacity. A small buffer results in fewer orders being buffered, increasing the probability of production teams finding an empty buffer. In that case

Algorithm 2 Two-Stage Timetabling Procedure

1: Initialize all parameters
2: Schedule the raw material gathering of the first i jobs, where $i = nrTeams$
3: **while** number of teams being $idle < nrTeams$ **do**
4: take first event e from $\mathcal{F}$, get time of event e and set $time = e.time$
5: **if** $teamStatus(i) <> waiting$ **then**
6: register end time of the operation
7: register the release of the equipment, depending on the resource used
8: **end if**
9:
10: **if** job j has pending operations and $i =$ buffer team **then**
11: **if** next stage is shortening **then**
12: reserve the machine ($nrShort = nrShort - 1$)
13: set end time of the event $e.time = time + p_{i,j}$, add $e.time$ to $\mathcal{F}$
14: $teamStatus(i) = busy$
15: **else if** job j finished the first stage **then**
16: place job j in buffer and increase the volume of b
17: **if** $b < bMax$ AND not all jobs in π have passed the first stage **then**
18: pick the next job $j + 1$ of π
19: set end time of the event $e.time = time + p_{i,j+1}$, add $e.time$ to $\mathcal{F}$
20: $teamStatus(i) = busy$
21: **else**
22: GoTo line 38 - Schedule complete job
23: **end if**
24: **end if**
25: **else if** job j has pending operations and $i <>$ buffer team **then**
26: **if** machine is available **then**
27: reserve the machine
28: set end time of the event $e.time = time + p_{i,j}$, add $e.time$ to $\mathcal{F}$
29: $teamStatus(i) = busy$
30: **else**
31: determine the earliest time a machine becomes available
32: set end time of the waiting event $e.time = time + w_{i,j}$, add $e.time$ to $\mathcal{F}$
33: $teamStatus(i) = waiting$
34: **end if**
35: **else if** job j has no pending operations **then**
36: register end time of j and calculate its delay
37: remove j from π
38: **if** $\pi <>$ empty **then**
39: pick the next job of π that satisfies the team size restrictions (i.e. $teamSize(i)$ is equal to the number of operators required) and determine the next pending operation
40: **if** machine is available **then**
41: reserve the machine
42: set end time of the event $e.time = time + p_{i,j}$, add $e.time$ to $\mathcal{F}$
43: $teamStatus(i) = busy$
44: **else**
45: determine the earliest time a machine becomes available
46: set end time of the waiting event $e.time = time + w_{i,j}$, add $e.time$ to $\mathcal{F}$
47: $teamStatus(i) = waiting$
48: **end if**
49: **else**
50: $teamStatus(i) = idle$
51: **end if**
52: **end if**
53: **end while**

only, the team is allowed to gather and shorten raw materials themselves. Note that the two-stage process is highly flexible as teams are allowed to pass to the other stage, however, a buffer team prioritizes operations in stage one whereas production teams prioritize operations in stage two. This flexibility is built in to prevent teams becoming idle, for example, waiting for orders in case of an empty buffer. In other terms, the two-stage flow model is focused on effective use of labor.

As a matter of fact, the determination of buffer size is a trade-off between throughput (equivalent to makespan) and tardiness. Assume, for example, a buffer of infinite size such that the buffer team solely performs operations at stage one. During actual production, the buffered volume grows steadily. At a certain point in time orders with a due date quite far from now are buffered as the team dedicated to stage one executes tasks very efficiently. However, capacity at stage two is lower compared to the QRM-based situation as only 4 teams are handling orders. They lack behind and tardiness will rise due to labor shortage. Processes, though, are efficient and man hours are saved by scrapping transfer time. At the end of the order list the buffer team will support at stage two and maximum capacity is reached accelerating the completion of jobs. Therefore, after this capacity increase, the teams are able to catch up and towards the end of the order list jobs will meet their due date again. In the end a good makespan performance will be observed, however, tardiness is expected to be high as during a certain amount of time (i.e. in the middle of the order list) capacity at stage two is insufficient to complete orders on-time.

In conclusion, the two-stage production process eliminates some of the inefficiencies of the current QRM-based process. Cornerstone of this flow model is still the QRM philosophy, but allows a limited number of orders to be buffered. Buffer size can be used as a parameter to deviate to a certain percentage from the current production process. In case the buffer size is set to 0, the current QRM-based process is modelled as the buffer is non-existent. Increasing the capacity will change the workload distributions between the buffer and production teams. Setting an unlimited buffer, the two stages are actually modelled as two separate departments. In chapter 8 performances under different buffer sizes are discussed in order to find the best fitting buffer capacity at Royal Dekker.

6.4 Model Validation and Verification

An important aspect in building simulation models is the verification and validation of the developed models with the aim of ensuring accurate and credible results. In this section both verification and validation of the flow-model are discussed.

6.5 Model Verification

Verification of a simulation model is the process of confirming the model is correctly implemented. In other terms, does the mathematical model work as we conceptually predicted. Verification should not be seen as a static process that has to be performed on some pre-determined times, it rather should be thought of as an iterative check of the model. We, therefore, used verification approaches to find and fix errors during the implementation of our model. However, stating all different steps taken to accomplish this will be beyond anyone's interest. Hence, only a short description of the verification process performed on the final model will be given.

To ensure the implemented model does as it is supposed to, the debugging mode was used allowing to take single steps through the code. Going through the code step by step made it able to verify each step taken. Besides, the output window of the simulation run was specifically designed to follow jobs through the process facilitating the verification process. A screenshot of the simulation output window can be found in Appendix D. It provides for each job the due date and shows all the individual operations including their start and end time. Besides, a team identification number, total waiting time and delay incurred by each job are shown, though not shown in the screenshot due to readability. We conclude that the simulation model is verified by analyzing its behaviour. In other terms, the model implicitly does what it is supposed to do.

6.6 Model Validation

Whereas verification concerns the relationship between a conceptual model and the simulation program, model validation checks the accuracy of the model's representation regarding the real system. A three-step validation approach is followed to assure our model meets its purpose by representing the facility of Royal Dekker:

1. Design a model with high face validity.
2. Validate model assumptions.
3. Determine the representative output of the model and compare with the real system output.

Face Validity
Face validity denotes experienced and knowledgeable experts of the real system examining the model and its output for reasonableness. In our case, both the production planner and one of the plant supervisors were involved during the development process and checked the outcomes to validate them with the actual observed performances. Both concluded the model is highly representative to the real world, although one of them pointed out the furlough of employees can influence the performances, while not being included in the model.

Model Assumptions
Let us first discuss assumptions made about the availability of data. In section 6.1.5 it is noted that all information about jobs is available at the start of the simulation. Although, this is actually true the timespan of 4 weeks is not. Due to the short delivery times promised by Royal Dekker, only jobs to be delivered within 5 working days are known. However, to come up with more reliable results the planning horizon was extended to 4 weeks. This does not by any means influence the validity of our model as it works perfectly fine for a shorter planning horizon. Moreover, the assumption was only made to use more data and hence, increase reliability of the outcomes.

Structural assumptions about the operation at the production facility were made in accordance with job shop literature. To increase the validity some assumptions of the classical JSP were dropped and replaced by extensions to the mathematical model, such as multi-resources, parallel machines, reentrant jobs, and partial blocking. Only two assumptions actually led the conceptual model deviate from reality. Firstly, operators are assumed to work 8 hours per day for 5 days a week without taking leave or being ill. Secondly, in contradiction with the assumptions that machines do not break down this is of course to happen in reality. On the other hand, machine availability (i.e. uptime) for the facility in The Hague was on average 98.5%[2] in 2019. So, the influence on performance indicators due to machine break downs is rare.

Output Validity
To determine the deviation in terms of performances found by the simulation model and those observed in reality a test run was performed. This included simulating the current situation at Royal Dekker. That is, the QRM-Based production process combined with a fixed team setting of 3-operator teams only. As will be explained later, jobs are currently scheduled by a simple earliest due date principle (discussed in section 7.1). Data set of Appendix C was used, which does represent all the demand incurred by Royal Dekker during June 2019 (in total 202 orders). Actual performances during this month are subtracted from the *Production Year Report 2019*, in which all measures are denoted per month. Actual total labor is calculated by adding the hours worked during normal operation time and hours during overtime minus furlough. Outcomes are depicted in Table 6.1. It can directly be seen the makespan prediction by our model is highly accurate, difference from reality is only 0.6%. An average accuracy is observed for the number of

[2] *Source: Production Year Report 2019*

tardy jobs. This can be explained, by the production planner making hands-on changes when a certain order is due while the model does not. In conclusion it can be said the model is valid with a high degree of accuracy.

Table 6.1: Output Validity on Test Run

	Total Labor	**Makespan**	**Tardy Jobs**	**On-time delivery %**
Real System	1867.3	155.6	77	61.9%
Simulation Model	1855.8	154.7	69	65.8%

7 Sequencing Heuristics

Although, small instances of the flow-shop problem can be solved to optimality by exact methods in polynomial time, they are not suitable for large-scale problems (Zhang et al., 2019). In the early days of JSP literature, Garey et al. (1976 showed that the job shop scheduling problem is NP-complete in case the number of machines grows beyond 2 ($m \geq 3$) while minimizing the makespan and NP-complete for $m \geq 2$ while minimizing the mean throughput time. As the multi-resource hybrid flow-shop with blocking is a generalization of the classical JSP it is certainly NP-Complete. Consequently, solving large problem instances within acceptable time requires approximation methods. Those approximation methods do not guarantee achieving exact solutions, however they do result in near optimal solutions, within moderate computing times. Three methods in particular were used to solve the problem under consideration. Firstly, the well-known earliest due date (EDD) rule is applied. In section 7.2 a hybrid dispatching rule algorithm is proposed based on results found in literature. Lastly, a genetic algorithm is build by fitting the best parameters to our problem in section 7.3.

7.1 EDD-Rule

Known to be one of the first approximation methods used to solve job shop problems, priority dispatching rules (PDRs) are frequently applied mainly due to their ease of implementation and their computational efficiency. A large variety of rules are mentioned in literature, all of them assigning a value to each waiting job according to some method and the job with the best score is selected to be scheduled next. A well-known and easy to implement PDR is the earliest due date rule. According to EDD the job with the earliest due date is scheduled first, the job with the next earliest due date is second, and so on. Hence, jobs are scheduled in increasing order of their delivery dates. Intuitively scheduling the job with the most critical due date next makes sense, and most of the human schedulers make use of this logic. Among many others, Baker (2005) stated that EDD is an effective approach in minimizing tardiness.

Currently, the planning of orders is based on two rules: 1) orders are scheduled based on their due date and 2) orders with the same due date are sorted according to their order number. That is to say, orders to be delivered on the same date are sequenced according the first-in-first-out (FIFO) principle. This logic will be applied to create an ordered sequencing list π that is used as an input parameter to the timetabling procedures discussed in chapter 6.

7.2 EDD-SPT Algorithm

The EDD is defined by counting the arrival and the agreed lead time, but does not take into account differences of processing times between jobs. A commonly applied PDR taking processing times into account is the shortest processing time (SPT) rule. SPT schedules the job with the smallest processing time first, the job with the next shortest processing time second, and so on. Hence, orders are scheduled in increasing order of their processing times. The SPT-rule has shown to dominate other PDRs with respect to optimizing the mean tardiness criterion (Haupt, 1989). The effectiveness of rules related to SPT was also shown by Chang et al. (1996). Another result obtained by these authors was the good performance of due date related rules in case the criteria are based on total tardiness.

Single priority dispatch rules, on the whole, deliver a bad job in minimizing the makespan. To solve the bi-objective problem introduced in this research a so-called hybrid algorithm is developed, in which both the EDD- and SPT-rule are used. Similar approaches are commonly mentioned in literature, where several rules are used simultaneously or next to each other to end up with a better result. Figure 7.1 shows the structure of the proposed sequencing algorithm. As it is clear from this figure, at first the order list is initialized and the array with team sizes s is generated. The list is then sorted based on the EDD-rule. All jobs with the earliest due date, denoted by the variable d, are selected. In case this selection does not include an order corresponding with the size of the available team a new selection is made of jobs with due date $d = d + 1$. As soon as this condition is met, the selected jobs with due date d are ordered according to the SPT-rule. Thereafter, the job first job satisfying the team size constraint is picked. The operations of this job are scheduled according to the procedure described in Algorithm 1. Once all jobs have been scheduled the algorithm is terminated.

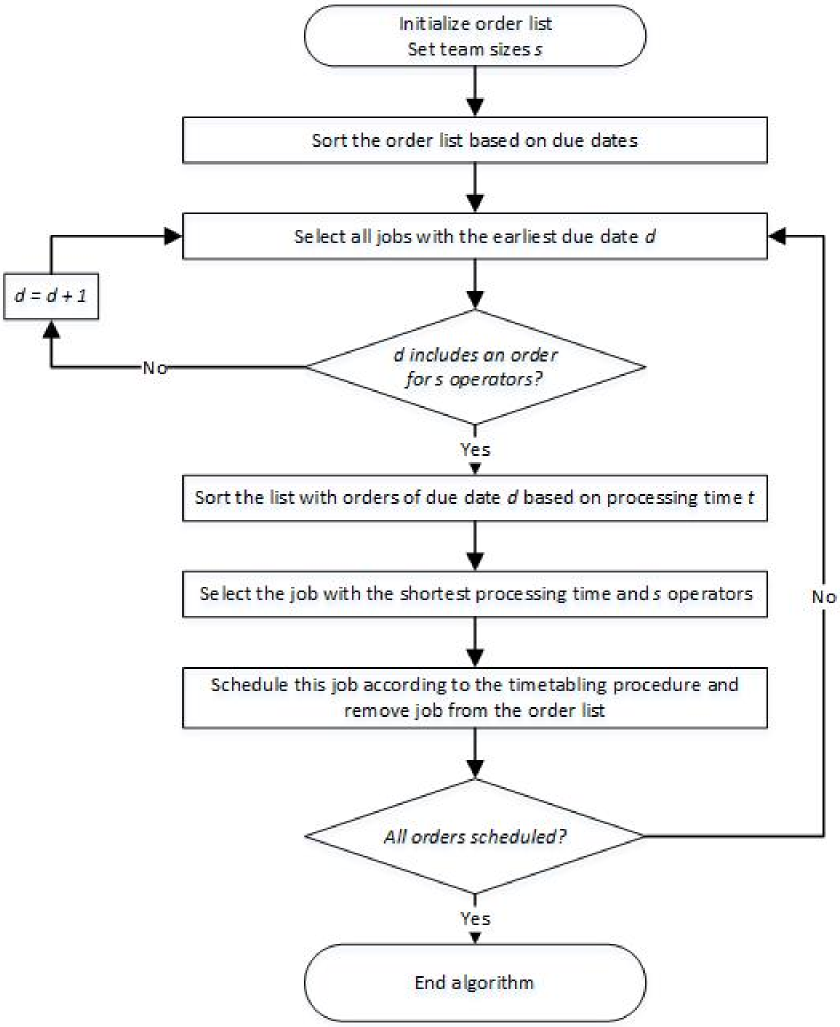

Figure 7.1: EDD-SPT Sequencing Algorithm

Based on a detailed analysis of the results obtained by scheduling jobs only according to their due date, the performance of the hybridized dispatching rule algorithm is believed to be better. Most of the delay turned out to be incurred by the last orders on congested days. This pattern returns every two or three days, on which multiple orders do not meet their requested delivery date. In most cases a bunch of orders are delayed, making the delay increase rapidly. Scheduling jobs based on processing time does not necessarily decrease the processing time but makes sure a smaller number of jobs are delayed at the end of a single day. This is due to the fact that small orders are processed at the beginning, only remaining with the jobs having a relatively long processing time. In other terms, fewer orders are delayed simultaneously resulting in the delay increasing less rapidly, not necessarily decreasing the total delay time of a particular delayed job. The suggested hybridized dispatching rule algorithm has been extensively tested on all 60 data sets for both the Fixed-3 and Fixed-2/3 methodologies. As for the single EDD scheduling algorithm discussed in Chapter 5, the results obtained for the total makespan, total delay and total number of delayed orders are discussed. As before, the significance level, alpha (α), is set to a value of 0.05, which is common in literature. In terms of computational effort the EDD-SPT algorithm, including the timetabling procedure, is highly efficient. The average computation time for a single run (i.e. 202 orders) was 4.6 seconds for the Fixed-3 setting and 5.2 seconds for the Fixed-2/3 setting. Without any doubt, the algorithm can be used during day-to-day operations as re-calculating the job sequence after adding a new job to the order list is almost negligible.

7.3 Genetic Algorithm

Experience has shown that dispatching is a relatively weak mechanism when used alone (Haskose et al., 2004). To overcome this burden a genetic algorithm (GA) was developed using the sequence found by the EDD-SPT algorithm as an initial solution. Moreover, genetic algorithms are based on the genetic evolution mechanism of biology. A population of candidate solutions to an optimization problem is evolved towards better solutions. The candidate solutions have a set of properties (called chromosomes) which can be mutated and altered. GA's have a global searching ability and can adjust search directions automatically and self-adaptively. First genetic algorithm applied to JSP was developed by Davis (1985), who used a GA as an indirect approach composing a preferred sequence of operations for every machine. Falkenauer and Bouffouix (1991) extended this method by encoding the operations to be processed on a machine as a preference string of symbols.

In order to apply GA to the scheduling problem under consideration, the chromosomes need to be encoded. Each chromosome represents a possible solution (i.e. sequence of jobs) to the problem. Before the actual execution of the genetic process starts an initial solution is required, which is based on the sequence found by the EDD-SPT sequencing algorithm. Each chromosome is evaluated based on two criteria, makespan and tardiness. To do so, a fitness function is proposed to calculate the fitness score of each individual. Selection of parent chromosomes is based on their fitness scores using a roulette wheel procedure. Crossover and mutation operators are applied to pairs of individuals to create offspring. Each of those steps is explained in more detail in the remainder of this section.

7.3.1 Chromosome Enconding

As mentioned above, the first task in applying a genetic algorithm consists of defining an encoding scheme to represent solutions of the problem. Generally, a distinction is made between direct and indirect encoding, the choice between them depends on the characteristics of the problem (Mati et al., 2011). Within direct encoding the representation scheme describes an actual solution, for example the list om machines and time slots that are used to perform the operations. The main disadvantage of this encoding stems from the fact that crossover operators would have to be substantially modified to generate feasible solutions taking precedence constraints into account (Falkenauer and Bouffouix, 1991). A more suitable, and widely used method in optimization problems, is the indirect representation scheme, where the chromosome contains information that is used to build a solution rather than providing a complete solution for the problem (Talbi, 2009).

To be able to use standard crossover operators indirect encoding is used. The representation is a permutation of n jobs. Each gene in the chromosome represents a job, whereas the chromosome indicate the priority of jobs (see Figure 7.2). In the scheme job i is the first to be scheduled according to the genetic algorithm and job k is the last, where job k is the n^{th} job. The actual schedule is build by the timetabling algorithm, which can be said to be a decoder.

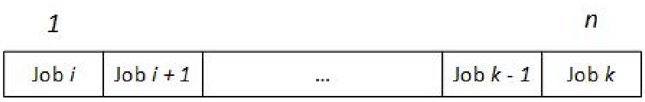

Figure 7.2: Permutation encoding scheme

7.3.2 Initial Solution

The performance of a genetic algorithm depends strongly on the initial population. Constructing a goods initial population, by applying simple heuristics, enables the GA to arrive at a (sub)-optimal solution more quickly (Etiler et al., 2004). As the sequences obtained by the EDD-SPT algorithm are the best up until now, it will be the basis from which the initial population is drawn.

The sequence provided by the algorithm is the first member of the initial solution. Remaining chromosomes of the initial population are generated by randomly swapping 5 pairs (i.e. 10 jobs) of genes of the first member. As a total number of 202 jobs have to be processed, swapping 5 pairs means a random modification of almost 10%. Main purpose of swapping those genes is to start with a variety of instances to prevent premature convergence. After conducting a multiple numerical experiments with population sizes of 10, 20 and 50 chromosomes, the size of the population was set to 20 chromosomes. This number returned on average the best results, although, in a few instances a population of 50 gave slightly better results it did not outweigh the long computation time. Based on our observations, this computation time is better deployed by increasing the number of generations with a population size of 20 individuals.

7.3.3 Chromosome Evaluation

Fitness of chromosomes is evaluated using the timetabling procedure. This procedure returns multiple values of which two in particular are used for optimization, namely makespan and total tardiness. The probability of an individual being selected as parent is directly proportional to its fitness value. That is, the higher the fitness score the larger the probability the individual creates offspring. Notice, however, in our case a high value of either total makespan or total tardiness is undesirable. The objective function values returned from the timetabling procedure are transformed into a fitness score by normalization. The normalized fitness value of an individual, f_{norm}, is calculated according to Equation 7.1, where denominator is equal to the sum of the makespan and total tardiness.

$$f_{norm} = \frac{100}{C_{max} + \sum_{j=1}^{n} T_j} \qquad \forall\ chromosomes \tag{7.1}$$

7.3.4 Parent Selection

As mentioned above, chromosomes from the current population are selected based on their fitness scores to form the mating pool. Purpose of parent selection is to keep fit individuals (solutions) and get rid of the bad ones from one generation to another. However, a high population diversity is desirable to prevent premature convergence. By keeping a diverse set of individuals a large part of the solution space can be explored, improving the quality of the solution (Mati et al., 2011). In this method, the widely used roulette wheel selection is applied to select parents.The roulette wheel is constructed as follows:

1. Calculate the total fitness value of the population:
   $F = \sum_{i=1}^{20} f_{norm}(i)$
2. Calculate the cumulative fitness c_i for each individual i:
   $c_i = \sum_{x=1}^{i} f_{norm}(x)$
3. Generate a random number r between 0 and F
4. Select the first individual i for which holds that $r < c_i$

Note that with the above method some individuals can be selected multiple times, whereas others are not selected at all. Those that are selected are added to a mating pool, to which crossover and mutation is applied with a certain probability. However, in our case the size of the mating pool (i.e. number of parents) is not equal to the population size. Next to the roulette wheel selection an elitist selection is applied. Elitism involves copying a small proportion of the fittest individuals, unchanged, into the next generation. The number of elite individuals should be carefully picked as selecting too many converges the problem to the nearest local maximum.

Efficiency of GA's depends to a high degree on the selection of control parameters, such as the number of elite individuals. Table 7.1 presents the results of extensive parameter testing. For each of the mentioned parameter settings the genetic algorithm was run 5 times, and the best results are depicted in Table 7.1. *Cprob* denotes the probability of crossover, *Mprob* is the probability of mutation, *# Elite* the number of elite individuals, and *# iter.* shows how many generations are created. For now, assume the parameter setting with *Cprob*= 0.8 and *Mprob*= 0.1 gives the best results. Fixing those parameter settings and varying the number of elite chromosomes between 0, 2, and 4 respectively it can directly be concluded that 2 elite individuals outperforms the others. Hence, the mating pool exists of 18 parents (population size - elite members).

Table 7.1: The best parameter sets for the benchmark problem (Fixed-3 basis data set)

Parameter	**Cprob**	**Mprob**	**# Elite**	**# iter.**	**Makespan**	**Delay**	**Fittest**	**Run (s)**
Cprob	0.7	0.05	2	50	155.12	83.43	0.4192	517
&	0.7	0.10	2	50	154.98	81.38	0.4231	529
Mprob	0.7	0.20	2	50	154.52	79.28	0.4277	513
Cprob	0.8	0.05	2	50	154.52	80.23	0.4256	553
&	0.8	0.10	2	50	154.78	76.35	0.4327	569
Mprob	0.8	0.20	2	50	154.72	77.95	0.4298	576
Cprob	0.9	0.05	2	50	154.05	77.62	0.4317	675
&	0.9	0.10	2	50	154.72	81.55	0.4233	684
Mprob	0.9	0.20	2	50	154.72	77.95	0.4298	681
# Elite	0.8	0.10	0	50	154.95	85.82	0.4153	613
	0.8	0.10	2	50	154.78	76.35	0.4327	569
	0.8	0.10	4	50	154.62	82.70	0.4214	567
# iterations	0.8	0.10	2	50	154.82	76.32	0.4327	569
	0.8	0.10	2	100	154.78	74.85	0.4355	812
	0.8	0.10	2	200	154.62	74.22	0.4370	1536

7.3.5 Crossover

Aim of the genetic search is to find the best permutation of the jobs. To do so, genetic operators are applied to parents creating new population members. One of the most significant phase in a GA is crossover, which is the partial exchange of information (genes) using a cross-interval chosen at random. Parents from the mating pool are chronologically selected from the mating pool, that is parent 1 is crossed with parent 2, parent 3 with parent 4 and so on.

Crossover is applied with a certain probability, most common in literature are 0.7, 0.8, and 0.9. Those settings have been tested on the basis data set and results can be found in Table 7.1. In all three instances (different values for *Mprob*) a change of 0.7 performs worse than either a probability equal to 0.8 or 0.9. Comparing those probability, they score equally well in the case of a mutation probability of 0.2. Based on dominance relation of the other two scenarios (i.e *Mprob* is 0.05 or 0.1) neither of the two values of *Cprob* can be chosen. However, the overall best fitness was found by setting *Cprob*= 0.8 and *Mprob*= 0.1 and, therefore, the probability of crossover in our genetic algorithm is set to 0.8.

Although, the crossover itself is a relatively ease procedure, the encoding scheme of scheduling problems, usually imply difficulties contrary to encoding with binary values. Swapping a random interval of jobs between parents create duplicates and missing jobs in the sequence of the children. Therefore, extra operations have to be added to the standard swapping procedure. A lot of popular crossover procedures exist in the literature regarding job shop scheduling. To name a few, Position

Based Crossover (PBX), Partially Mapped Crossover (PMX), and Linear Order Crossover (LOX). This genetic algorithm uses the LOX procedure that is developed by Falkenauer and Bouffouix (1991) as it is shown to be one of the best performers among the others (Etiler and Toklu, 2001). As the procedure is rather complex, a short description is given below, but we refer to the article of the developers for a clear (and graphical) explanation. The LOX procedure works as follows:

1. Pick the first unscheduled pair of parents chronologically from the mating pool.
2. Choose a crossover interval by randomly selecting two cut points.
3. Determine the genes (i.e. jobs) of both parents within the interval and remove those from both parents.
4. Remove from parent 1 the jobs to be implemented from parent to and vice versa, leaving some "holes" in both parents sequences.
5. For both parents shift the remaining jobs, filling all the empty slots, such that the crossover interval is completely empty.
6. Insert crossover genes of parent 1 into the empty slots of parent two and vice versa.

7.3.6 Mutation

Mutation is crucial in finding good quality solutions as it maintains the diversity in the population and averts premature convergence. However, mutation of chromosomes should be rare to not let the genetic algorithm shift away from good solutions. From Table 7.1 and the explanation in section 7.3.5 the mutation probability was fixed to 0.1. In this research simple exchange mutation was used by randomly swapping two genes in the same parent.

7.3.7 Stopping Criterion

As the problem proposed is NP-hard, no optimal solution can be pursued and used as a stopping criterion. The number of generations (iterations) will, therefore, be used as the termination criterion. Deciding on the number of iterations is some kind of a bifurcation, whereas a low number of generations decreases the probability of finding the best result, but on the other a too high number of iterations results in extensive computation times. In the literature a wide range of values are used, some only generate 20 new populations whereas others chose to use 1000 or even more iterations. This number is highly dependent on the problem complexity.

As the bi-objective multi-resource hybrid flowshop with partial blocking can be marked as rather complex tests where only performed with 50, 100, and 200 iterations respectively. Results are shown in Table 7.1, depicting an increase in fitness each time the number of iterations is increased. The increase in fitness, though, from 100 to 200 iterations is very limited when in fact the computation time increases dramatically from 812 to 1536 seconds. Besides only in 2 out of the 5 test runs the stopping criterion of 200 gave better results. One of the goals of this research is to find a scheduling procedure that can be used on day-to-day basis, therefore, the GA is terminated once it reaches 100 iterations. We are aware that by using this number of generations not always the best (sub)-optimal solution is found, but iterating 200 times found better results in only the minority of test runs.

8 Analysis and Results

To determine the effectiveness of the production methodologies in combination with the proposed heuristics analyses have been performed on multiple randomly generated data sets. For each data set the makespan, total delay, number of delayed orders and total waiting time have been measured.

To be able to draw conclusions on the output variables the central limit theorem (CLT) was used. This theorem states that the distribution of sample means approximate a normal distribution as the sample size becomes larger, assuming that all samples are identical in size. As a general rule of thumb, sample sizes equal to or greater than 30 are deemed sufficient for the CLT to hold. Although extremely time-consuming a total of 60 samples have been randomly generated and tested for all methodologies (twice the minimum number of samples needed). The procedure for generating the data sets is as follows:

1. Take the data set of 202 orders, ordered at their arrival date (FIFO) as a basis.
2. Randomly generate a due date on an interval from 1 to 20 for each job.
3. Sort the jobs according to their due dates

Note that neither the number of jobs nor the service rates are changed. Keeping the number of jobs constant throughout all the 60 data sets allows us to use the central limit theorem. Secondly, service rates are kept equal because both, the available data is not sufficient to make reasonable and well-founded assumptions about the service rate (and corresponding distribution) and because determining those rates would be a study on its own as explained in chapter 4. All 60 data sets do therefore have the same workload, but differ in terms of due dates.

For all measured performance indicators minimum and maximum values, averages, standard deviations and confidence intervals are given. Let x_i denote the indicator of interest and n the number of data sets, then the following formulas were used:

$\overline{x}$	average of the measured variable ($\overline{x} = \frac{\sum x}{n}$)
x_{min}	minimum of the measured variable ($x_{min} = min\{x_1, x_2, ...x_n\}$)
x_{max}	maximum of the measured variable ($x_{max} = max\{x_1, x_2, ...x_n\}$)
s^2	(sample) variance of the measured variable ($s^2 = \frac{\sum(x_i - \overline{x})^2}{n-1}$)
s	(sample) standard deviation of the measured variable ($s = \sqrt{s^2}$)

Often confidence intervals centered around the the estimator $\overline{x}$ are used to give an estimate that expresses in some sense the spread of the variable. As for the significance level, alpha (α), a value of 0.05 is chosen, which is common in literature. Hence, with 95% confidence we can assert that the actual value x lies within the corresponding interval. In order to obtain the 95% confidence interval, we use the following equation:

$$I = [\overline{x} - z_{1-\alpha/2} * \frac{s}{\sqrt{n}}, \overline{x} + z_{1-\alpha/2} * \frac{s}{\sqrt{n}}] \quad (8.1)$$

Results obtained for the makespan, total tardiness and total number of delayed orders are discussed in great detail in sections 8.1 to 8.4. Although the waiting time is an interesting value, indicating whether or not Royal Dekker needs to invest in adding extra equipment, it does not by any means indicate the performance of a certain methodology. As waiting times rise, total makespan and total delay are negatively affected. Hence, waiting time can be seen as a secondary performance measure and therefore is only discussed briefly.

Table 8.1 provides an overview of all configurations tested. First of all, different team settings are used for each scenario. Those team settings have been extensively discussed in section 5.2. In all configurations, except one, the setting with teams of 3 operators only is included as a benchmark.

This team composition, however, could not be tested for the fourth scenario since this entails the two-stage production process in which at least one team composed of 2 employees is required to perform operations at stage one. Secondly, two kinds of timetabling procedures (see chapter 6) are incorporated in the analysis. That are both, the QRM-based production process presenting the current order processing at Royal Dekker and the Two-Stage production process imposed to eliminate the inefficiencies of the first. Lastly, different sequencing methods, discussed in chapter 7, are added to each test configuration. Each of the three proposed scheduling methods is tested for the QRM-based production process, providing an answer to the second research question. The two-stage production process is only modelled with the EDD-SPT algorithm to obtain whether or not better results are obtained compared to the QRM-based process to answer the third research question. Section 8.5 concludes this chapter and provides an answer to both research questions.

Table 8.1: Test configurations

Section	Team Methodologies	Timetabling Method	Sequencing Method
8.1	Fixed-3, Fixed-2/3, Variable-2/3	QRM-Based	EDD-rule
8.2	Fixed-3, Fixed-2/3	QRM-Based	EDD-SPT Algorithm
8.3	Fixed-3, Fixed-2/3	QRM-Based	Genetic Algorithm
8.4	Fixed-2/3, Variable-2/3	Two-Stage	EDD-SPT Algorithm

8.1 QRM-Based Results with EDD Rule

First scenario discussed is the combination of QRM-based timetabling (Algrithm 1) with sequence optimization by earliest due date (section 7.1). This methodology reflects the current situation at Royal Dekker in case the Fixed-3 approach is employed. Aim of this simulation is to evaluate the results of the proposed improvements related to the production process taking the restrictions of QRM into account. Outcome of the analysis is used to answer the first research question: *How to improve the current QRM-based production process keeping the necessary elements to guarantee short delivery times?*

As a first test the proposed production methodologies (Fixed-3, Fixed-2/3, Variable-2/3) were tested on data sets representing the demand on a single week, i.e. the basis data set of 20 work-days was split into four individual weeks. The obtained results were very inconclusive, none of the methodologies seemed to dominate the others. However, in two cases the Fixed-2/3 showed the lowest makespan and the lowest number of delayed orders compared to the other approaches. Testing the methods on the actual data set of 202 orders the Fixed-2/3 approach performed best on total makespan, whereas the Variable-2/3 approach performed best on total delay and number of orders delayed. These results are just primarily and the actual analysis was performed on 60 randomly generated data sets. Summarized results for each of the performance indicators can be found below, whereas detailed results can be found in Appendix E.

Makespan
Total makespan represents the time taken in order to finish all production requests. In other terms, it denotes the time elapsed until the last team finishes. Results for the total makespan, measured in hours, are depicted in Table 8.2. One would expect to come across little variation in makespan as the workload of all data sets is equal. This certainly holds true for both the Fixed-3 and Fixed-2/3 methodologies, with a standard deviation of 1.3 and 1.2 respectively. The standard deviation of the third approach is considerably larger, however, the coefficient of variation is 2.7%, indicating that standard deviation is just 2.7% of the average makespan.

Based on the average of 60 test runs the Fixed-2/3 approach performs best. When considering the 95% confidence interval this methodology seems to dominate the others as the upper bound, 144.1 hours, is less than the lower bounds of the Fixed-3 and Variable-2/3 methods, 153.8 and

148.4 hours respectively. Looking at the detailed results (Appendix C) the Fixed-3 methodology in all cases requires more time to finish all orders than the Fixed-2/3 approach. Besides, only in 6 out of the 60 runs the Variable-2/3 approach outperforms the Fixed-2/3 production strategy. It, therefore, can be stated that the Fixed-2/3 strategy strictly dominates the Fixed-3 approach, whereas it weakly dominates the Variable-2/3 approach.

Table 8.2: Results for the total makespan (in hours) for QRM-Based EDD scenario

Team	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-3	151.4	157.4	1.3	154.2	153.8	154.5
Fixed-2/3	141.8	146.0	1.2	143.8	143.5	144.1
Variable-2/3	141.7	159.4	4.0	149.4	148.4	150.5

Tardiness

Results for the total tardiness, measured in hours, can be found in Table 8.3. The total delay is simply the sum of all delays encountered by individual orders. Standard deviations of all methodologies are large, even up to 80% of the average. This large variance can be explained as follows; in case the process is in control delays will be low or even non-existing, however, once the processing of orders stays behind delays for upcoming orders keep on rising and become extremely high. It is noteworthy the only production approach to encounter zero delay, although being a single case, was the Fixed-2/3 strategy.

Again, the Fixed-2/3 approach outperforms the other methodologies on both average total delay and confidence interval. Although, the upper bound of this approach is less than the lower bounds of the other methods both the Fixed-3 and Variable-2/3 methodology performed better in some runs and are therefore only weakly dominated by the Fixed-2/3 approach.

Table 8.3: Results for the total delay (in hours) for QRM-Based EDD scenario

Team	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-3	17.6	731.8	181.0	257.5	211.7	303.3
Fixed-2/3	0.0	372.4	93.7	116.4	92.7	140.1
Variable-2/3	5.2	576.3	121.8	174.3	143.4	205.1

Number of Delayed Orders

Delay expressed in time is a very popular optimization criterion and is widely used in literature. It, however, says little about the number of customers receiving a late delivery while this is an important figure for Royal Dekker's management due to the fines received by customers in case of delayed delivery (most of the time not depending on the actual delay time). To get more insight into this problem results for the number of delayed orders are shown in Table 8.4. The results are in line with those of the delay measured in hours. An important result is the encountered decrease in delayed orders compared to the current situation in case the Fixed-2/3 strategy will be used.

Table 8.4: Results for the number of delayed orders for QRM-Based EDD scenario

Team	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-3	6	134	34.8	72.0	63.2	80.8
Fixed-2/3	0	73	17.7	32.1	27.6	36.6
Variable-2/3	4	101	21.4	42.7	37.3	48.1

Waiting Time

Waiting time of teams is costly and should be prevented. In certain occasions, however, waiting appears to increase other performances. For example, scheduling an almost due job right before the original start time of another job may result in the latter waiting, but on the other hand reduces the tardiness of the first job. That reasoning may hold for the Fixed-2/3 approach, which is subjected to a reasonably higher waiting time than the other approaches, while all other performance indicators are in favour of this method.

On the other hand, assuming waiting time is incurred proportionally over all teams, on average an individual team has to wait for around 1 hour. This once again shows the machinery available is to a great extent sufficient to process the incurred demand. Whereas, the actual bottleneck is related to the number of teams or equivalent the available amount of labor.

Table 8.5: Results for the waiting time (in hours) for QRM-Based EDD scenario

Team	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-3	0.2	7.5	1.7	3.3	2.9	3.8
Fixed-2/3	2.1	12.8	2.0	5.5	5.0	6.0
Variable-2/3	1.6	11.7	2.0	4.8	4.3	5.3

Sensitivity of the Methodologies
As mentioned earlier, both methods with mixed teams are sensitive to the ratio of orders handleable by either 2 or 3 operators. The base data set, and thereby all the 60 test sets exists for 53% of orders handleable by 2 operators and 47% of the orders require 3 operators. Based on data from the past two years the ratio can be assumed to be 50/50. However, in case this ratio will shift over time the chosen production methodology may not be the best anymore. To indicate how the performance of the approaches will differ once the ratio will shift two tests were done:

- Ratio 70/30: constraints about the weight limit of a product where loosened to create a data set in which 70% of the orders could be handled by 2 operators.
- Ratio 30/70: constraints about the weight limit of a product where strengthened to create a data set in which 30% of the orders could be handled by 2 operators.

Test results are shown in Table 8.6. It is obvious that the Fixed-3 approach is not sensitive to any change in ratio as the team size is always fixed to 3 employees. The Fixed-2/3 performs worst on both ratios 70/30 and 30/70, which was expected as the mix of teams was primarily build for the 50/50-ratio. Contrariwise, the Variable-2/3 approach outperforms all others in case the percentage of 2-operators orders increase.

Table 8.6: Comparison for different order ratios for QRM-Based EDD scenario

	Fixed-3			Fixed-2/3			Variable-2/3		
Ratio 2/3	Makesp.	Delay	# del.	Makesp.	Delay	# del.	Makesp.	Delay	# del.
53/47	154.2	52.5	22	145.4	21.8	11	153.0	16.7	6
70/30	154.2	52.5	22	181.0	1561.1	115	143.9	1.7	1
30/70	154.2	52.5	22	208.5	3159.4	134	208.5	3159.4	134

Conclusion
Main goal of this analysis was to answer the first research question: *How to improve the current QRM-based production process keeping the necessary elements to guarantee short delivery times?*. Based on all three parameters, total makespan, total delay and number of orders delayed it can be concluded that the Fixed-2/3 production methodology outperforms all other approaches. Although this method is sensitive in the short term to the ratio of 2- and 3-operator orders it leads by far to the best results in the long term. To answer the research question, the current QRM-based production process can be improved by implementing the Fixed-2/3 production method. This entails working with two teams of 3 operators and three teams of 2 operators simultaneously.

Implementing such a strategy will on average result in downsizing the makespan by 7% and both the total delay and number of delayed orders will be more than halved.

8.2 QRM-Based Results with EDD-SPT Algorithm

Main focus of the previous analysis was on improving the current production process, rather than finding a more efficient sequencing method. This section builds on the results achieved in the previous analysis. Results obtained by introducing a fixed combination of teams with 2 and 3 operators are further improved by proposing a more sophisticated algorithm combining the earliest due date and shortest processing time rules. Moreover, this scheduling algorithm is applied to the Fixed-2/3 methodology, as it seemed to outperform other approaches, and to the current situation, all teams consisting of 3 operators, used as a benchmark to indicate the improvement. Since the previous section emphasized the Variable-2/3 approach stayed behind, it was not incorporated in further analysis. Detailed results can be found in Appendix F.

Makespan
Results obtained for the total makespan are shown in Table 8.7. It is noteworthy that the standard deviation has increased dramatically compared to using the EDD-rule as a single decision criterion. Especially for the scenario with combined team sizes, the hybridized algorithm negatively affects all makespan parameters. On average the total completion time increases by 0.5% and 0.4% for the Fixed-3 and Fixed-2/3 approaches respectively. In comparison with the results in Table 8.2 for the Fixed-2/3 approach, maximum makespan encountered within the data set rose from 146.0 to 149.1 hours. This can mainly be explained by the lack of sensitivity of SPT to shop load variations (Haupt, 1989). Besides, dispatching rules related to processing times are mainly focused on decreasing the mean or total tardiness, rather than minimizing completion times. As an unwanted side effect, the end of the simulation (i.e. last jobs of the data set) is stretched since the last jobs have the longest processing times. Because of this, the idle time of certain teams at the end of the simulation run is larger compared to the single use of EDD.

Although the results for both the combined team size methodology and the fixed 3-operator setting deteriorated, the dominance relation stays untouched. That is to say, for all instances the makespan obtained by the Fixed-2/3 approach is less than the ones achieved by setting the team size fixed to 3 operators.

Table 8.7: Results for the total makespan (in hours) for QRM-Based SPT-EDD scenario

Team	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-3	152.7	158.3	1.4	155.0	154.6	155.3
Fixed-2/3	141.9	149.1	1.7	144.5	144.0	144.9

Tardiness
Contrary to the effect seen in terms of completion time, the total tardiness is positively influenced by the EDD-SPT scheduling algorithm. From Table 8.8 it can directly be concluded that on average the delay is decreased by 21.0% and 15.5%, for the Fixed-3 and Fixed-2/3 settings respectively. Again the fixed 3-operator approach benefits more from the hybridization. The 95% confidence interval for both settings is narrowed by almost 24%, by which the relevance of the lower and upper bound increases.

Table 8.8: Results for the total delay (in hours) for QRM-Based SPT-EDD scenario

Team	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-3	15.5	545.9	131.9	203.4	170.0	236.8
Fixed-2/3	1.6	291.6	71.5	98.4	80.3	116.5

Number of Delayed Orders
Even better results are obtained regarding the number of delayed orders compared to the improvement percentages of the total delay. Table 8.9 denotes the parameters regarding the actual number of orders being delayed. The average number of postponed deliveries is decreased by 26.7% and 19.3%, for the Fixed-3 and Fixed-2/3 settings respectively.

On average a total number of 25.9 orders were delayed using the Fixed-2/3 setting combined with the EDD-SPT algorithm. Comparing this value to the benchmark (i.e. the Fixed-3 setting with EDD-rule) the improvement is extraordinary. Initially, the average number of orders with postponed delivery was 72 (see Table 8.4), indicating an improvement of 64.0%. Referring to the actual problem of Royal Dekker, a total amount of 385.000 € has been paid in fines in one year. By applying the proposed algorithms to the multi-resource flexible flow-shop little over 246.500 € is "saved", as fewer fines will be incurred.

Table 8.9: Results for the number of delayed orders for QRM-Based SPT-EDD scenario

Team	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-3	5	111	25.4	53.5	47.0	59.9
Fixed-2/3	1	56	13.4	25.9	22.5	29.3

Waiting Time
Just like the performance indicators, total waiting time has decreased by using the hybridized dispatching rule algorithm. In comparison with using solely EDD to determine the sequence of jobs decreases of 24.2% and 11.3% are observed for the Fixed-3 and Fixed-2/3 approach respectively.

Table 8.10: Results for the waiting time (in hours) for QRM-Based SPT-EDD scenario

Team	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-3	0.3	6.9	1.2	2.5	2.2	2.9
Fixed-2/3	1.6	10.1	2.0	4.9	4.4	5.4

Conclusion
In this subsection, we analyzed results for scheduling orders by the EDD-SPT algorithm for the present QRM-based production process. To simulate this scenario orders were first sequenced according to the algorithm depicted in Figure 7.1. Thereafter, the QRM-based timetabling procedure described by Algorithm 1 was used to simulate the actual execution of jobs throughout the production facility. Results for both the Fixed-3 and Fixed-2/3 team settings were discussed.

Overall the results observed by using the hybridized EDD-SPT algorithm proofed to be better than those obtained by using the EDD rule (section 8.1) on its own. A minor increase in makespan was observed for both the Fixed-3 and Fixed-2/3 methodology. However, total tardiness decreased up to 21%, as well as the total number of tardy jobs. Main conclusion is that, in accordance with results obtained in literature, the hybridized EDD-SPT algorithm outperforms the EDD-rule.

8.3 QRM-Based Results with Genetic Algorithm

Although results obtained by the EDD-SPT have shown to improve the current performance of the flow-shop at Royal Dekker significantly, dispatching rules in general provide poor results compared to other more sophisticated techniques. In this respect, the performance of the genetic algorithm proposed in section 7.3 will be analyzed in more detail below. The job sequences found for each of the 60 data sets by the EDD-SPT algorithm are used to form the initial population.

For each problem instance the genetic algorithm was executed once with a stopping criterion of 100 generations. A GA not necessarily returns the same results each time it is executed due to its randomness. Therefore it could easily be possible to observe better results by running the GA a few times on the same problem instance. This, however, is time consuming and not realistic as in practice the scheduler will probably run the GA only once. For detailed results see Appendix G.

Makespan

After running the GA for all 60 data sets a minor improvement compared to the EDD-SPT algorithm was found regarding the makespan. Table 8.11 provides an overview of the results found. For the Fixed-3 and Fixed-2/3 team settings improvements of 0.8% and 1.0% were found respectively. Although this may seem as negligible in terms of labor costs it is quite an effective outcome. In absolute terms the makespan was reduced by 1.2 hours for the fixed 3-operator team setting and by 1.4 hours for the flexible 2- and 3-operator setting. Since 12 employees are employed by the company a total saving in labor of respectively 14.6 and 16.8 hours can be achieved.

Table 8.11: Results for the total makespan (in hours) for QRM-Based GA scenario

Team	**Min.**	**Max.**	**Std. Dev.**	**Avg.**	**Lower bound**	**Upper bound**
Fixed-3	151.6	157.2	1.4	153.8	153.4	154.1
Fixed-2/3	140.8	147.4	1.5	143.1	142.7	143.5

Tardiness

Looking at Table 8.12 the results for total tardiness found by the genetic algorithm are significantly better than those obtained by the EDD-SPT algorithm. A reduction of 11.1% and 13.4% is observed for the Fixed-3 and Fixed-2/3 setting respectively. More importantly, compared to the current situation (see Fixed-3 setting in Table 8.3) the improvement is equal to 29.8% and 66.9% respectively. Besides, the maximum tardiness of all 60 problem instances (i.e. 245.2 hours) is still below the average total delay perceived in the current situation. It can easily be said the genetic algorithm combined with the flexible team setting of 2- and 3-operator teams shows excellent results. A more thorough comparison with the current situation is provided in section 8.5, in which the same conclusion will be drawn for all performance indicators.

Table 8.12: Results for the total delay (in hours) for QRM-Based GA scenario

Team	**Min.**	**Max.**	**Std. Dev.**	**Avg.**	**Lower bound**	**Upper bound**
Fixed-3	11.0	525.6	121.5	180.8	150.1	211.6
Fixed-2/3	1.0	245.2	62.0	85.3	69.6	100.9

Computational Performance

All algorithms were coded in the programming language of Excel, Microsoft Visual Basics for Applications (VBA) and run on an HP Zbook with an Intel Core i7 processor. Despite VBA is by far an efficient programming language, it was chosen because of its direct connection to Excel. Royal Dekker can directly subtract data from their ERP system to Excel. Besides most (traditional) manufacturing companies do not have access to Matlab or Java.

In terms of computation time the genetic algorithm is extremely costly compared to the EDD-SPT algorithm. Table 8.13 depicts the average time required to 1) determine the sequence and 2) timetable the jobs according to the QRM-based timetabling procedure. Although, the GA's computation time can be seen as a disadvantage one should also take into account the size of the actual problem. In less around 16 minutes some performance indicators are improved by over 65% for a scheduling horizon of 4 weeks. This magnitude of computation time is well acceptable for daily use and is found very reasonable by the production planner at Royal Dekker.

Table 8.13: Comparison of computational performance (in seconds)

Team setting	**EDD-SPT**	**GA**
Fixed-3	4.6	530.9
Fixed-2/3	5.2	975.8

8.4 Two-Stage Based Results with EDD-SPT Algorithm

Previous three test configurations focused on improving performance indicators while keeping the current production process to a great extent similar to the current situation at Royal Dekker. In this section, the aim is to determine whether a two-stage flow model outperforms the current QRM-based production process. To do so, the timetabling procedure shown in Algorithm 2 is simulated together with the EDD-SPT algorithm (Figure 7.1).

As explained earlier a team setting solely composed of teams with three operators is not suitable for this configuration as at least one team of 2 operators is required at stage one. Furthermore, although not incorporated in the previous two analysis due to bad performance the Variable-2/3 team setting is considered to deliver better performances in combination with the Two-Stage flow model. This expectation is mainly due to the workload balancing capabilities of this team setup. Taking all the time savings mentioned in section 6.3 into account, the workload for each stage was calculated. Table 8.14 shows for each stage and team size the total workload and workload per team respectively. It can easily be observed that there is a lack of capacity at stage two to process jobs requiring 2 operators only. With a workload of 122.3 hours, the buffer team finishes earlier than the 2-operator production teams at stage two with a workload of 182.4 hours. However, in case the maximum buffer size is reached or once all orders passed through stage one, the buffer team supports by processing jobs at stage two. Nevertheless, teams composed of 3 employees are supposed to finish earlier than their counterparts with 2 operators. This will eventually result in a major loss of labor due to the idleness of 3-operator teams in case a Fixed-2/3 team setting is used. A solution to the problem sketched above is found in the use of a more variable team setting, in which 3-operator teams support 2-operator teams by also processing jobs that can be processed by 2 operators. This Variable-2/3 approach was explained in detail in section 5.2.

Table 8.14: Workload Balance in Two-Stage Flow Model

Stage	**Order Type**	**# Teams**	**Total Workload**	**Workload/team**
One	2-operators	1	122.3	122.3
Two	2-operators	2	364.8	182.4
Two	3-operators	2	200.2	100.1

Earlier it was already stated that buffer capacity impacts both makespan as well as total tardiness. One should be careful by just setting the buffer size as large as possible as it may negatively affect the performances. Realistic lower and upper bounds were chosen to the buffer size and simulation runs were performed varying the capacity within those bounds. Average volume of a single order within the date set is slightly above $2.5m^3$. Minimum buffer size was therefore set to $6m^3$, such that around 2 orders can be buffered. Two factors, in particular, determine the upper bound of the buffer. Firstly, the actual floor space at Royal Dekker is limited. Secondly, by increasing the volume allowed to be buffered the number of orders at the physical buffer location rises. Too many orders result in production teams searching for their order and thus deteriorating the time saving gained by the Two-Stage production environment. Taking the above into consideration, a maximum was set to $20m^3$. Hence, upper bound of the buffer is equal to 8 averagely sized jobs. For running the simulation buffer size was increased by steps of $2m^3$.

The remainder of this section discusses all three performance indicators making use of histograms to visualize the effect on a specific performance measure when varying the buffer size. Detailed test results can be found in Appendix H.

Makespan

Throughput, and thereby total makespan, is assumed to decrease up to a certain level by increasing the buffer size. From Figure 8.1 it can be seen this certainly holds for the variable team setting, but does not for the Fixed-2/3 approach. The latter can be blamed on the behavior of teams throughout the actual processing of jobs. A small buffer also leads to an increased number of jobs processed by the buffer team at stage two. Assume the buffer is set to the lowest volume ($6m^3$) it can easily be imagined the buffer is full most of the time. Whenever the buffer passed its maximum volume the dedicated buffer team starts processing orders at stage two until there is sufficient buffer space available. Besides, teams with 3 operators are subjected to a greater chance to gather raw materials themselves since there the probability of an order requiring at least 3 operators being buffered is relatively low. Hence, workload at stage two is more balanced, positively effecting the makespan. Contrariwise, a large buffer size results in the buffer team processing more jobs at stage one, including gathering raw materials for teams with 3 operators. Eventually, this leads to a larger workload for the two production teams with 2 operators and thus a larger makespan.

Makespan seems to decrease up to a buffer size of $18m^3$ for the Variable-2/3 team setting. Contrary to the fixed team setting workload balanced is build in. As expected, this feature seems to improve the performance within a Two-Stage flow model whereas it did not in the current QRM-based flow model. Average makespan, of all 60 problem instances, was found to be 145.0 hours with a buffer size of $18m^3$ and 145.2 hours in case the buffer capacity was either fixed to $16m^3$ or $20m^3$.

Overall, the variable team configuration outperforms the fixed setting for all buffer sizes. An optimum was found by setting the maximum capacity to $18m^3$, although, performance differs only slightly compared to adjacent buffer sizes. The 95% confidence interval of the optimum solution is (144.4 - 145.6) which is a little more spread out than intervals of (near) optimal solutions found for the QRM-based process.

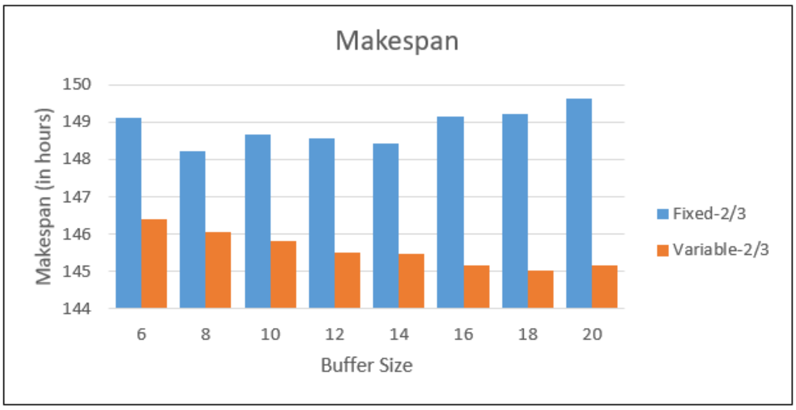

Figure 8.1: Effect of buffer size on makespan

Tardiness
Results obtained for total tardiness are depicted in Figure 8.2. Same effects observed for total makespan are found for the delay time. The same reasoning used above holds for the effects seen here. Besides, as already explained in section 6.3, from a certain buffer size tardiness is assumed to increase due to a lack of capacity at stage two. In other terms, orders stay in the buffer for a long period of time without being further processed even in case they are already due.

Dominance of the variable team setting observed for the makespan is even greater for the tardiness. Whereas the minimum tardiness by using the Fixed-2/3 approach is equal to 86.9 hours ($b = 6$), the variable setting returned a minimum delay of 47.9 hours ($b = 18$). Note that the difference is almost a factor 2. Again it can be said the variable team setting with a buffer size of $18m^3$ provides the best solution. Corresponding confidence interval is (37.2 - 58.5) which is narrow compared to results found to the QRM-based flow model, increasing the predictability and decreasing the variance during day-to-day operations.

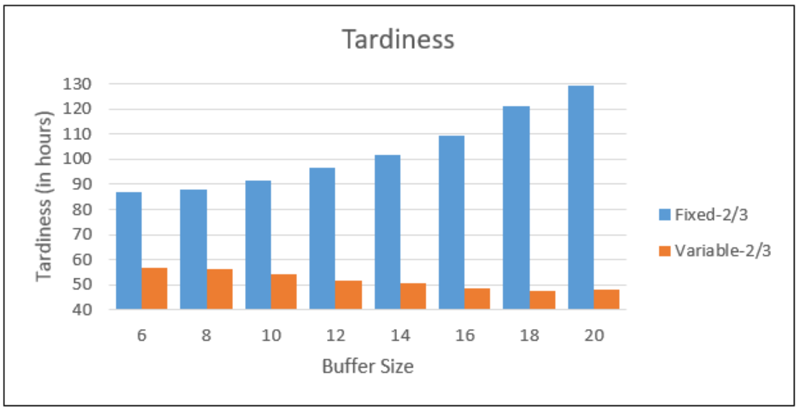

Figure 8.2: Effect of buffer size on total tardiness

8.5 Conclusion

In this chapter results for different configurations have been extensively analyzed and discussed. Aim of this chapter is to provide an answer to which method performs best in terms of both makespan and total tardiness. Table 8.15 shows the combined results of all configurations tested. Mentioned multiple times already, the current situation at Royal Dekker is depicted by the QRM-based timetabling procedure with a fixed setting of 3-operator teams only, while using the EDD-rule to sequence jobs. This scenario is denoted as our benchmark to determine the relative performances of other test configurations. Left-handed side of Table 8.15 describes the settings of each individual test case. In the middle, best observed (average) values for each configuration are given. The last two columns show the (percent based) difference to the benchmark situation for makespan and tardiness respectively.

Table 8.15: Comparison of average performances with current situation

Configuration			*Performances*		*Difference with benchmark*	
Timetabling	**Sequencing**	**Team setting**	**Makespan**	**Tardiness**	**Makespan**	**Tardiness**
QRM-Based	EDD-Rule	Fixed-3	154.2	257.5	-	-
QRM-Based	EDD-Rule	Fixed-2/3	143.8	116.4	-6.7%	-54.8%
QRM-Based	EDD-Rule	Variable-2/3	149.4	174.3	-3.1%	-32.3%
QRM-Based	EDD-SPT	Fixed-3	155.0	203.4	+0.5%	-21.0%
QRM-Based	EDD-SPT	Fixed-2/3	144.5	98.4	-6.3%	-61.8%
QRM-Based	GA	Fixed-3	153.8	180.8	-0.3%	-29.8%
QRM-Based	GA	Fixed-2/3	143.1	85.3	-7.2%	-66.9%
Two-Stage	EDD-SPT	Fixed-2/3	148.2	86.9	-3.9%	-66.3%
Two-Stage	EDD-SPT	Variable-2/3	145.0	47.9	-6.0%	-81.4%

From the table it can directly be concluded that none of the configurations strictly dominate the others. In other terms, no single methods performs best on both makespan and total delay. Best in terms of makespan is the QRM-based production environment with a fixed team setting of both 2- and 3-operator teams optimized by the genetic algorithm. A makespan of 143.1 hours is observed, which is an improvement of 7.2% with respect to the current situation. We should, however, mention that the same configuration with a sequence optimized by EDD-rule leads to a makespan nearly as good as the GA, namely 143.8 hours. As the computation time is considerably less, only a few seconds whereas the GA takes on average 976 seconds, this optimization method could be preferred during day-to-day production in case one is only interested in minimizing makespan.

However, the Fixed-2/3 team setting optimized by GA is the best overall configuration in case

Royal Dekker would hold on to the current QRM-based production process. Next to saving 7.2% in terms of production time, tardiness will be reduced by 66.9%.

Taking the Two-Stage production process into consideration, it results in a staggering reduction of 81.4% regarding the total delay time. This method can be said to be the best in terms of reducing tardiness. As explained in section 6.3 introducing a buffer results in a trade-off between makespan (throughput) and tardiness. As a matter of fact, that allegation seems to hold observing a makespan of 145.0 hours. Compared to the lowest makespan achieved, 143.1, almost 2 more hours are required by splitting the process into two stages. However, total tardiness is almost halved compared to the configuration with the best achieved makespan.

9 Conclusions and Recommendations

In this chapter, a conclusion is given by answering the formulated research questions. Moreover, in section 9.2 recommendations regarding both the interpretation of the outcomes and actual implementation of the proposed methods are discussed. Main limitations of the research are mentioned in section 9.3. Lastly, a few words about future research directions are provided.

9.1 Conclusions

The main objective of this research was to improve the performances on both makespan and tardiness in a hybrid multi-resource flow shop with partial blocking. Some intermediate conclusions have been drawn throughout the report, but no final answers have been given. In this section answers will be presented systematically for all three research questions.

1. How to improve the current QRM-based production process keeping the necessary elements to guarantee short delivery times?
Restrictions argued by Royal Dekker regarding team-based manufacturing and the Quick Response Manufacturing philosophy limited the search space dramatically. A solution was sought for in finding an optimal size for the production teams. Three team compositions were mentioned, Fixed-3, Fixed-2/3 and Variable-2/3 differing in terms of team size and flexibility. Afterward, all three methodologies were tested keeping the other parameters fixed to the current situation.

Both the fixed and variable team configuration with a mixture of 2- and 3-operator teams showed improvements to the current method. Particularly, the fixed composition with teams of 2 and 3 employees proofed to be very effective. Improvements of 6.7% and 54.8% were achieved for total makespan and total tardiness respectively. The current QRM-based production process can, therefore, be significantly improved by splitting the current teams into two teams composed of 3 operators and three teams composed of 2 operators. By doing so, more than half of the orders currently being delayed will be delivered on-time ensuring short delivery times.

2. How should production orders be scheduled such that costs are minimized respecting both the fixed on-time delivery percentage and the QRM-based production process?
To answer the second research question three sequencing heuristics have been proposed in section 7. First, a simple priority dispatching rule based on earliest due dates was discussed. Thereafter, based on findings in literature, a hybridized algorithm was developed combining both EDD and SPT. Lastly, a more sophisticated genetic algorithm was presented.

Results in Table 8.15 show the dominance of the GA combined with the fixed team composition found to be the best in research question 1 for a QRM-based environment. More precisely, improvements of 7.2% and 66.9% were found for total makespan and total tardiness respectively. The first, leads to a cost reduction as labor is saved, whereas the latter reduces the amount paid in fines. Referring to the actual problem of Royal Dekker, a total amount of 385.000 € has been paid in fines in one year. By applying the proposed algorithms to the multi-resource flexible flow-shop little over 257.500 € is "saved", as fewer fines will be incurred

3. Can performances be further improved when loosening some QRM-based restrictions creating a partial QRM-based process?
The main aim of the first two research questions is to improve the current production process without deviating too much from it. That is to say, both restrictions working in teams, as well as a single team executing all operations of an individual job, had to be respected. The third research question, though, has to provide an answer whether the current QRM-based production process is actually better in terms of makespan and tardiness compared to a Two-Stage flow model with limited buffering capabilities.

Following the reasoning of section 8.5, the QRM-based process, in fact, provides better results with respect to throughput (makespan). The best result obtained regarding total completion time with the EDD-SPT algorithm in combination with QRM-based timetabling was a reduction of 6.3%, whereas a decrease of 6.0% was observed with the Two-Stage timetabling procedure. However, in terms of tardiness reduction improvements of 61.9% and 81.4% were achieved for QRM-based and Two-Stage manufacturing respectively, clearly showing the effectiveness of allocating a buffer.

In conclusion, the Two-Stage flow model halves the total tardiness compared to the QRM-based flow model making use of the same EDD-SPT heuristic to sequence jobs. Although, this production model is subjected to a very minor increase in makespan (0.5 hours) the performance in terms of tardiness reduction makes it the best among all other configurations. Answering the research question, performances are further improved by loosening some QRM-based restrictions.

9.2 Recommendations

In this section an indication will be given how Royal Dekker can benefit from the proposed methods in this thesis. This both includes a piece of advice regarding improving the current QRM-based processes and a more extreme modification of current processes.

First of all, in case Royal Dekker sticks to the current QRM-based production process we highly recommend switching to a more flexible team composition. Best would be to adopt the fixed composition of two teams with three operators and three teams with two operators. One should, however, take its sensitivity to the ratio of orders handleable by either 2 or 3 operators into account. In case the ratio changes beyond certain limits, it can be wise to review whether the Fixed-2/3 still performs best or one should switch to a Variable-2/3 approach.

Secondly, our research showed a great improvement in performances by using a genetic algorithm. We recommend implementing this optimization heuristics in day-to-day scheduling. Implementation would be relatively easy as the programming language used is directly linked to Excel, which in turn connects with Royal Dekker's ERP-system.

Lastly, analysis of a Two-Stage flow model with a buffer capacity of $18m^3$ showed excellent results regarding total tardiness. Our advice is to implement this production strategy. This, though, would include partially deviating from the QRM philosophy. It would be wise to first start with buffering jobs and start working with a dedicated buffer team afterwards.

9.3 Main Limitations

Because of the assumptions made and the scope selected for our research, there exist some limitations that should be mentioned before acting upon the recommendations. Although we tried to present a realistically as possible flow model, a number of assumptions had to be made in order to deal with the mathematical side of our problem. Some issues may occur when implementing the presented solutions in daily operations. Besides, a solution was fitted based on a data set of 202 orders which is assumed to reflect the average demand (based on a Poisson process). When the arrival rate changes the conclusions of this research may no longer hold. Lastly, process disruptions such as machine breakdowns were not taken into consideration. These simplifications do create a gap between practice and the ideal scenario, probably affecting the performances negatively.

9.4 Further Research

This research provided an extensive analysis of multiple solution methodologies. Promising results are thought to be found by scheduling jobs using a genetic algorithm for the Two-Stage flow model. More research regarding this scenario is required to see whether performances can be further improved. Besides, to overcome the burden regarding changes to arrival rates benchmark problems from literature should be solved to see how the proposed heuristics handle different scenarios.

References

Abdul-Nour, G., Drodet, J., and Lambert, S. (1999). Mixed Production, Flexibility and SME. *Computers and Industrial Engineering*, 37(1):429–432.

Abyaneh, S. and Zandieh, M. (2012). Bi-objective hybrid flow shop scheduling with sequence-dependent setup times and limited buffers. *The International Journal of Advanced Manufacturing Technology*, 58(1):309–325.

Ahonen, H. and De Alvarenga, A. (2017). Scheduling Flexible Flow Shop with Recirculation and Machine Sequence-Dependent Processing Times: Formulation and Solution Procedures. *The International Journal of Advanced Manufacturing Technology*, 89(1):765–777.

Alvarez-Valdes, A., Fuertes, A., Tamarit, J., Gimenez, G., and Ramos, R. (2005). A heuristic to schedule flexible job-shop in a glass factory. *European Journal of Operational Research*, 165(2):525–534.

Baker, K. (2005). *Elements of Sequencing and Scheduling*. Tuck School of Bussiness, Hannover.

Cachon, G. and Terwiesch, C. (2013). *Matching Supply with Demand*. McGraw-Hill, New York.

Chang, Y., Sueyoshi, T., and Sullivan, R. (1996). Ranking Dispatching Rules by Data Envelopment Analysis in a Job-Shop Environment. *IIE Transactions*, 28(8):631–642.

Dauzere-Peres, S., Roux, W., and Lasserre, J. (1998). Multi-Resource Shop Scheduling with Resource Flexibility. *European Journal of Operational Research*, 107(2):289–305.

Davis, L. (1985). *Job-Shop Scheduling with Genetic Algorithm*, pages 136–140. Lawrence Erlbaum, Pittsburg, PA, USA.

Elmuti, D. (1997). Self-managed work teams approach: creative management tool or a fad? *Management Decision*, 35(3):233–239.

Etiler, O. and Toklu, B. (2001). Comparison of Genetic Crossover Operators Used in Scheduling Problems. *Gazi University Journal of Science*, 14(1):21–32.

Etiler, O., Toklu, B., Atak, M., and Wilson, J. (2004). A Genetic Algorithm for Flow Shop Scheduling Problems. *Journal of the Operational Research Society*, 55(8):830–835.

Falkenauer, E. and Bouffouix, S. (1991). *A Genetic Algorithm for the Job-Shop*. Proceedings of the IEEE International Conference on Robotics and Automation.

Garey, M., Johnson, D., and Sethi, R. (1976). The Complexity of Flowshop and Job Shop Scheduling. *Mathematics of Operations Research*, 1(2):117–129.

Haskose, A., Kingsman, B., and Worthington, D. (2004). Performance Analysis of Make-to-Order Manufacturing Systems under Different Workload Control Regimes. *International Journal of Production Economics*, 90(2):169–186.

Haupt, R. (1989). A Survey of Priority Rule Based Scheduling. *OR Spektrum*, 11(1):3–16.

Hecker, F., Hussein, W., Paquet-Durand, O., Hussein, M., and Becker, T. (2013). A case study on using evolutionary algorithms to optimize bakery production planning. *Expert Systems with Applications*, 40(17):6837–6847.

Lenstra, J. and Rinnooy Kan, A. (1979). Computational Complexity of Discrete Optimization Problems. *Annals of Discrete Mathematics*, 4(1):121–140.

Mati, Y., Lahlou, C., and Dauzere-Peres, S. (2011). Modelling and Solving a Practical Flexible Job-Shop Scheduling Problem with Blocking Constraints. *International Journal of Production Research*, 49(8):2169–2182.

Suri, R. (2010). *It's about time.* Taylor and Francis Inc., New York.

Talbi, E. (2009). *Metaheuristics: From Design to Implementation.* John Wiley & Sons, New Jersey.

Vouros, G. and Papadopoulos, H. (1998). Buffer Allocation in Unreliable Production Lines using a Knowledge Based System. *Computers and Operations Research*, 25(1):1055–1067.

Wang, H. (2005). Flexible Flow Shop Scheduling: Optimum, Heuristics and Artificial Intelligence Solutions. *Expert Systems*, 22(2):78–85.

Wang, L., Zhang, L., and Zheng, D. (2006). An effective hybrid genetic algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research*, 33(10):2960–2971.

Zhang, J., Ding, G., Zou, Y., Qin, S., and Fu, J. (2019). Review of Job Shop Scheduling Research and its New Perspectives Under Industry 4.0. *Journal of Intelligent Manufacturing*, 30(4):1809–1830.

Appendix A - Order Flows

A thorough analysis of all orders totally or partially handled by the production facility in The Hague has led to Table .1. The data has been subtracted from the ERP system of Royal Dekker. Data of 2018 applies to the complete year (i.e. January up to and including December), whereas data of 2019 applies to the months January up to and including September. After analysing the data subtracted from the ERP system it should be mentioned that both path F and H need to be deleted from the analysis. Both paths are physically not possible and can be identified as an administrative mistake.

Table .1: Overview of order flows and their frequency

Path	Path	# in 2018	# in 2019
A	Gathering → Sawing	31	31
B	Gathering → Planing	406	274
C	Gathering → Shortening → Sawing	4	8
D	Gathering → Shortening → Planing	74	112
E	Gathering → Sawing → Planing	666	493
F	Gathering → Sawing → Vianen	2	2
G	Gathering → Plann g → Vianen	169	102
H	Gathering → Shortening → Sawing → Vianen	0	1
I	Gathering → Sawing → Planing → Vianen	300	250
J	Gathering → Shortening → Planing → Vianen	26	33
K	Gathering → Shortening → Sawing → Planing	200	218
L	Gathering → Shortening → Sawing → Planing → Vianen	37	67
Total		**1915**	**1591**

Appendix B - Process Parameters

Table .2: Overview of process parameters for a team of size 3

Task Number	Description	Unit	Setup Time (min)	Processing Speed (unit/min)
DH-BE-0010	Gathering: Quality 1	m3	0	0.07
DH-BE-0020	Gathering: Quality 2	m3	0	0.07
DH-KO-0005	Shortening: variable length	m3	5	0.17
DH-KO-0010	Shortening: fixed length	m3	5	0.17
DH-ZA-K010	Sawing: 1 cut (2 parts)	m1	16	14
DH-ZA-K020	Sawing: 2 cuts (3 parts)	m1	16	7
DH-ZA-K030	Sawing: 3 cuts (4 parts)	m1	16	5
DH-ZA-K040	Sawing: 4 cuts (5 parts)	m1	16	4
DH-ZA-P010	Sawing: 1 cut (2 parts)	m1	16	14
DH-ZA-P020	Sawing: 2 cuts (3 parts)	m1	16	7
DH-ZA-P030	Sawing: 3 cuts (4 parts)	m1	16	5
DH-ZA-P040	Sawing: 4 cuts (5 parts)	m1	16	4
DH-ZA-S010	Sawing: 1 cut (2 parts)	m1	16	17
DH-ZA-S020	Sawing: 2 cuts (3 parts)	m1	16	17
DH-ZA-S030	Sawing: 3 cuts (4 parts)	m1	16	17
DH-ZA-S040	Sawing: 4 cuts (5 parts)	m1	16	17
DH-SC-0010	Planing: rough (1 part)	m1	12	18
DH-SC-0030	Planing: profile (1 part)	m1	16	17
DH-SC-0040	Planing: profile (2 parts)	m1	16	14
DH-SC-0060	Planing: profile (3 parts)	m1	16	14
DH-SC-0060	Planing: profile (4 parts)	m1	16	14

Table .3: Overview of process parameters for a team of size 2

Task Number	Description	Unit	Setup Time (min)	Processing Speed (unit/min)
DH-BE-0010	Gathering: Quality 1	m3	0	0.07
DH-BE-0020	Gathering: Quality 2	m3	0	0.07
DH-KO-0005	Shortening: variable length	m3	5	0.17
DH-KO-0010	Shortening: fixed length	m3	5	0.17
DH-ZA-K010	Sawing: 1 cut (2 parts)	m1	16	12
DH-ZA-P010	Sawing: 1 cut (2 parts)	m1	16	12
DH-ZA-S010	Sawing: 1 cut (2 parts)	m1	16	14
DH-SC-0010	Planing: rough (1 part)	m1	12	14
DH-SC-0030	Planing: profile (1 part)	m1	16	14
DH-SC-0040	Planing: profile (2 parts)	m1	16	12

Appendix C - Data Set

This Appendix contains all the detailed information regarding the data used during simulation and analysis.

Order nr.	Due Day	Team Size	Task (Gathering)	m3	Task (Short)	m3	Task (Saw 1)	m1	Task (Saw 2)	m1	Task (Plan)	m1
PVO19002830	1	2	DH-BE-0010	2,9			DH-ZA-P010	572,4			DH-SC-0030	1144,8
PVO19003150	1	3	DH-BE-0020	4,0	DH-KO-0005	4,0					DH-SC-0060	2280,0
PVO19003245	1	2	DH-BE-0010	1,2			DH-ZA-K010	462,0			DH-SC-0030	924,0
PVO19003403	1	3	DH-BE-0020	10,7			DH-ZA-K010	1582,0			DH-SC-0030	3164,0
PVO19003434	1	3	DH-BE-0020	0,3			DH-ZA-S030	122,5			DH-SC-0030	490,0
PVO19003435	1	2	DH-BE-0020	2,7			DH-ZA-S010	427,7			DH-SC-0040	855,4
PVO19003441	1	3	DH-BE-0020	0,8			DH-ZA-K020	151,0	DH-ZA-P010	453,0	DH-SC-0040	906,0
PVO19003480	1	3	DH-BE-0020	7,9							DH-SC-0030	2130,0
PVO19003241	2	3	DH-BE-0020	2,1			DH-ZA-S020	611,2			DH-SC-0030	1833,6
PVO19003265	2	2	DH-BE-0020	3,0			DH-ZA-K010	420,0	DH-ZA-P010	840,0	DH-SC-0040	1680,0
PVO19003319	2	3	DH-BE-0020	14,0	DH-KO-0010	14,0					DH-SC-0030	3256,0
PVO19003321	2	2	DH-BE-0020	2,1			DH-ZA-S010	496,7			DH-SC-0040	993,4
PVO19003001	3	2	DH-BE-0020	1,0			DH-ZA-S010	624,8			DH-SC-0040	1249,5
PVO19003002	3	3	DH-BE-0020	1,3			DH-ZA-P020	484,6			DH-SC-0040	1453,7
PVO19003152	3	3	DH-BE-0020	5,9	DH-KO-0005	5,9					DH-SC-0060	1688,5
PVO19003259	3	3	DH-BE-0010	5,0	DH-KO-0005	5,0					DH-SC-0030	1036,5
PVO19003291	3	2	DH-BE-0020	0,8							DH-SC-0040	298,9
PVO19003294	3	2	DH-BE-0020	1,9			DH-ZA-S010	717,6			DH-SC-0040	1435,2
PVO19003297	3	2	DH-BE-0020	2,0							DH-SC-0030	2028,6
PVO19003328	3	2	DH-BE-0010	3,4	DH-KO-0005	3,4	DH-ZA-K010	830,5			DH-SC-0040	1661,0
PVO19003353	3	3	DH-BE-0010	6,4			DH-ZA-K010	1180,4			DH-SC-0030	2360,7
PVO19003465	3	2	DH-BE-0010	0,9	DH-KO-0005	0,9	DH-ZA-S010	413,7			DH-SC-0030	827,3
PVO19003507	3	3	DH-BE-0020	2,8							DH-SC-0030	690,0
PVO19002820	4	3	DH-BE-0020	10,5	DH-KO-0005	10,5					DH-SC-0060	5493,0
PVO19003075	4	2	DH-BE-0020	8,4	DH-KO-0005	8,4	DH-ZA-S010	2268,0			DH-SC-0030	4536,0
PVO19003256	4	2	DH-BE-0020	4,5	DH-KO-0005	4,5	DH-ZA-K010	1252,0			DH-SC-0040	2504,0
PVO19003320	4	2	DH-BE-0010	0,8							DH-SC-0030	921,3
PVO19003341	4	2	DH-BE-0020	0,8							DH-SC-0010	298,9
PVO19003342	4	3	DH-BE-0020	0,3			DH-ZA-P030	46,8			DH-SC-0030	187,2
PVO19003343	4	2	DH-BE-0020	1,4							DH-SC-0030	803,6
PVO19003350	4	3	DH-BE-0020	0,8			DH-ZA-S030	148,8			DH-SC-0040	595,2
PVO19003352	4	3	DH-BE-0010	5,1	DH-KO-0005	5,1	DH-ZA-K010	1105,5			DH-SC-0030	2211,0
PVO19003151	5	3	DH-BE-0010	4,4							DH-SC-0060	2179,0
PVO19003154	5	3	DH-BE-0010	1,7	DH-KO-0005	1,7					DH-SC-0060	1320,0

Order nr.	Due Day	Team Size	Task (Gathering)	m3	Task (Short)	m3	Task (Saw 1)	m1	Task (Saw 2)	m1	Task (Plan)	m1
PVO19003178	5	2	DH-BE-0020	0,6							DH-SC-0030	648,6
PVO19003246	5	2	DH-BE-0010	4,8	DH-KO-0005	4,8	DH-ZA-S010	927,2			DH-SC-0030	1854,4
PVO19003247	5	2	DH-BE-0010	6,9							DH-SC-0030	2620,0
PVO19003248	5	2	DH-BE-0010	3,1							DH-SC-0030	1171,2
PVO19003464	5	2	DH-BE-0020	1,0			DH-ZA-S010	319,2			DH-SC-0030	638,4
PVO19003471	5	3	DH-BE-0020	0,5							DH-SC-0030	133,4
PVO19003508	5	2	DH-BE-0020	0,6	DH-KO-0005	0,6	DH-ZA-K010	276,8			DH-SC-0030	553,6
PVO19002722	6	3	DH-BE-0010	10,3			DH-ZA-K030	317,7				
PVO19003180	6	2	DH-BE-0020	0,4			DH-ZA-S010	152,5			DH-SC-0030	305,0
PVO19003181	6	2	DH-BE-0020	0,5			DH-ZA-P010	94,9			DH-SC-0030	189,8
PVO19003201	6	2	DH-BE-0020	0,2			DH-ZA-S010	93,6			DH-SC-0040	187,2
PVO19003202	6	2	DH-BE-0020	0,3	DH-KO-0005	0,3	DH-ZA-S010	98,9			DH-SC-0040	197,8
PVO19003281	6	3	DH-BE-0010	0,9	DH-KO-0005	0,9					DH-SC-0050	197,1
PVO19003290	6	2	DH-BE-0020	0,4	DH-KO-0010	0,4	DH-ZA-S010	170,5			DH-SC-0030	341,0
PVO19003293	6	3	DH-BE-0020	0,8			DH-ZA-P030	152,5			DH-SC-0030	610,0
PVO19003396	6	3	DH-BE-0020	1,3			DH-ZA-P020	484,6			DH-SC-0040	1453,7
PVO19003397	7	2	DH-BE-0010	3,1			DH-ZA-S010	908,3			DH-SC-0040	1816,6
PVO19003402	7	2	DH-BE-0020	4,6			DH-ZA-S010	1372,0			DH-SC-0040	2744,0
PVO19003404	7	2	DH-BE-0020	0,7			DH-ZA-S010	218,4			DH-SC-0040	436,8
PVO9003406	7	2	DH-BE-0010	3,6			DH-ZA-K010	1113,0			DH-SC-0040	2226,0
PVO19003463	7	2	DH-BE-0010	1,6	DH-KO-0005	1,6	DH-ZA-P010	251,6			DH-SC-0030	503,1
PVO19003405	8	2	DH-BE-0020	1,3			DH-ZA-S010	717,6			DH-SC-0040	1435,2
PVO19003414	8	2	DH-BE-0020	1,4							DH-SC-0040	1600,8
PVO19003415	8	3	DH-BE-0020	1,0			DH-ZA-S020	456,4			DH-SC-0030	912,7
PVO19003416	8	2	DH-BE-0020	2,0							DH-SC-0030	2028,6
PVO19003428	8	2	DH-BE-0020	0,6			DH-ZA-K010	313,5			DH-SC-0040	627,0
PVO19003429	8	3	DH-BE-0020	1,6							DH-SC-0010	630,3
PVO19003470	8	2	DH-BE-0010	1,1	DH-KO-0005	1,1					DH-SC-0030	363,4
PVO19003509	8	2	DH-BE-0010	0,9	DH-KO-0005	0,9					DH-SC-0030	266,6
PVO19003510	8	3	DH-BE-0010	5,7	DH-KO-0005	5,7					DH-SC-0050	1368,9
PVO19003065	9	2	DH-BE-0020	4,8			DH-ZA-K010	1805,0			DH-SC-0040	3610,0
PVO19003068	9	2	DH-BE-0020	1,0			DH-ZA-S010	319,2			DH-SC-0030	638,4
PVO19003082	9	2	DH-BE-0010	2,0							DH-SC-0030	1069,2
PVO19003106	9	2	DH-BE-0020	4,0			DH-ZA-S010	1280,8			DH-SC-0030	2561,5

Order nr.	Due Day	Team Size	Task (Gathering)	m3	Task (Short)	m3	Task (Saw 1)	m1	Task (Saw 2)	m1	Task (Plan)	m1
PVO19003407	9	2	DH-BE-0020	1,8			DH-ZA-S010	671,0			DH-SC-0030	1342,0
PVO19003430	9	3	DH-BE-0020	0,7	DH-KO-0005	0,7	DH-ZA-P020	198,0			DH-SC-0030	594,0
PVO19003431	9	2	DH-BE-0020	2,4	DH-KO-0010	2,4					DH-SC-0030	478,5
PVO19003466	9	3	DH-BE-0010	3,7							DH-SC-0010	733,5
PVO19003506	9	3	DH-BE-0010	1,4	DH-KO-0005	1,4	DH-ZA-S010	198,0			DH-SC-0050	396,0
PVO19003550	9	2	DH-BE-0010	1,7	DH-KO-0005	1,7					DH-SC-0040	588,6
PVO19003575	9	2	DH-BE-0020	3,5	DH-KO-0005	3,5	DH-ZA-S010	1398,7			DH-SC-0030	2797,4
PVO19003129	10	2	DH-BE-0020	0,6			DH-ZA-K010	66,0	DH-ZA-P010	132,0	DH-SC-0030	264,0
PVO19003408	10	2	DH-BE-0020	2,0			DH-ZA-S010	1234,3			DH-SC-0040	2468,6
PVO19003482	10	3	DH-BE-0010	2,8			DH-ZA-K010	1471,5			DH-SC-0060	2943,0
PVO19003484	10	3	DH-BE-0020	1,4							DH-SC-0030	292,5
PVO19003485	10	2	DH-BE-0020	2,3	DH-KO-0010	2,3	DH-ZA-S010	866,9			DH-SC-0040	1733,8
PVO19003486	10	2	DH-BE-0020	0,3							DH-SC-0030	71,2
PVO19003545	10	3	DH-BE-0010	1,4	DH-KO-0005	1,4	DH-ZA-K010	550,2			DH-SC-0060	1100,4
PVO19003546	10	3	DH-BE-0020	0,9			DH-ZA-K020	213,2			DH-SC-0040	639,5
PVO19003574	10	2	DH-BE-0020	0,4			DH-ZA-S010	188,7			DH-SC-0040	377,3
PVO19003576	10	3	DH-BE-0020	3,8			DH-ZA-S010	1473,7			DH-SC-0050	2947,4
PVO19003616	10	2	DH-BE-0020	0,6	DH-KO-0005	0,6	DH-ZA-K010	292,4			DH-SC-0030	584,8
PVO19003624	10	2	DH-BE-0010	0,5							DH-SC-0030	332,8
PVO19003686	10	2	DH-BE-0010	0,9	DH-KO-0005	0,9	DH-ZA-S010	413,7			DH-SC-0030	827,3
PVO19003558	11	2	DH-BE-0020	2,9			DH-ZA-S010	1495,0			DH-SC-0040	2990,0
PVO19003559	11	3	DH-BE-0020	0,2			DH-ZA-K010	36,0	DH-ZA-S020	72,0	DH-SC-0030	216,0
PVO19003578	11	2	DH-BE-0020	2,3			DH-ZA-S010	1283,4			DH-SC-0040	2566,8
PVO19003599	11	3	DH-BE-0020	1,0							DH-SC-0030	271,2
PVO19003600	11	2	DH-BE-0020	2,6							DH-SC-0030	1524,0
PVO19003625	11	2	DH-BE-0020	2,7			DH-ZA-S010	427,7			DH-SC-0040	855,4
PVO19003626	11	3	DH-BE-0020	0,1			DH-ZA-K020	24,5	DH-ZA-P010	73,5	DH-SC-0040	147,0
PVO19003648	11	2	DH-BE-0020	4,5			DH-ZA-K010	1698,7			DH-SC-0040	3397,4
PVO19003649	11	3	DH-BE-0020	0,4			DH-ZA-K010	70,8	DH-ZA-S020	141,6	DH-SC-0030	424,8
PVO19003153	12	2	DH-BE-0020	0,4			DH-ZA-P010	93,4			DH-SC-0040	186,8
PVO19003183	12	3	DH-BE-0020	1,5							DH-SC-0060	919,7
PVO19003199	12	2	DH-BE-0020	3,3			DH-ZA-K010	908,4			DH-SC-0040	1816,8
PVO19003200	12	3	DH-BE-0020	2,7			DH-ZA-P020	765,4			DH-SC-0030	2296,2
PVO19003295	12	3	DH-BE-0020	1,6							DH-SC-0030	253,7

Order nr.	Due Day	Team Size	Task (Gathering)	m3	Task (Short)	m3	Task (Saw 1)	m1	Task (Saw 2)	m1	Task (Plan)	m1
PVO19003440	12	3	DH-BE-0020	1,4			DH-ZA-S020	519,0			DH-SC-0040	1557,0
PVO19003547	12	2	DH-BE-0020	2,2			DH-ZA-S010	1362,5			DH-SC-0030	2725,0
PVO19003556	12	2	DH-BE-0020	3,7			DH-ZA-S010	2396,6			DH-SC-0040	4793,2
PVO19003595	12	2	DH-BE-0020	1,1			DH-ZA-K010	309,6			DH-SC-0040	619,2
PVO19003690	12	3	DH-BE-0020	1,7							DH-SC-0050	825,1
PVO19003691	12	2	DH-BE-0020	1,2							DH-SC-0040	749,7
PVO19001528	13	3	DH-BE-0010	1,2			DH-ZA-S020	360,8			DH-SC-0030	1082,4
PVO19002888	13	3	DH-BE-0020	4,1	DH-KO-0010	4,1	DH-ZA-P040	299,3			DH-SC-0030	1496,5
PVO19002978	13	3	DH-BE-0020	2,8							DH-SC-0030	690,0
PVO19003078	13	2	DH-BE-0020	2,6			DH-ZA-S010	1435,2			DH-SC-0040	2870,4
PVO19003131	13	3	DH-BE-0020	0,7							DH-SC-0030	192,0
PVO19003156	13	3	DH-BE-0020	3,2			DH-ZA-K010	801,1			DH-SC-0030	1602,2
PVO19003182	13	3	DH-BE-0020	1,8	DH-KO-0005	1,8	DH-ZA-S020	687,0			DH-SC-0030	2061,0
PVO19003204	13	3	DH-BE-0020	0,2			DH-ZA-P020	44,0			DH-SC-0030	132,0
PVO19003647	13	3	DH-BE-0010	0,3			DH-ZA-K010	121,0			DH-SC-0070	242,0
PVO19003666	13	2	DH-BE-0020	1,2			DH-ZA-S010	474,7			DH-SC-0040	949,3
PVO19003673	13	3	DH-BE-0020	1,0							DH-SC-0030	306,2
PVO19003694	13	3	DH-BE-0020	2,0			DH-ZA-S010	770,0			DH-SC-0050	1540,0
PVO19003734	13	3	DH-BE-0020	0,2	DH-KO-0010	0,2	DH-ZA-S020	60,5			DH-SC-0030	181,5
PVO19000954	14	2	DH-BE-0020	1,0			DH-ZA-S010	623,5			DH-SC-0040	1247,0
PVO19002256	14	3	DH-BE-0010	4,0							DH-SC-0010	742,5
PVO19002710	14	3	DH-BE-0020	8,0							DH-SC-0030	2148,6
PVO19003037	14	3	DH-BE-0020	0,7			DH-ZA-S020	248,5			DH-SC-0030	496,9
PVO19003627	14	3	DH-BE-0020	1,1			DH-ZA-K020	214,2	DH-ZA-S010	642,6	DH-SC-0040	1085,2
PVO19003628	14	3	DH-BE-0020	1,4							DH-SC-0030	556,2
PVO19003665	14	3	DH-BE-0020	2,5							DH-SC-0030	670,3
PVO19003671	14	2	DH-BE-0020	4,1			DH-ZA-K010	1545,7			DH-SC-0040	3091,3
PVO19003674	14	2	DH-BE-0020	0,4			DH-ZA-P010	96,0			DH-SC-0030	192,0
PVO19003676	14	3	DH-BE-0010	15,8			DH-ZA-K010	1676,2				
PVO19003678	14	3	DH-BE-0010	1,4	DH-KO-0010	1,4	DH-ZA-K010	442,8			DH-SC-0060	885,6
PVO19003714	14	3	DH-BE-0010	0,7	DH-KO-0010	0,7					DH-SC-0060	316,1
PVO19003778	14	3	DH-BE-0020	6,5							DH-SC-0010	2513,7
PVO19002780	15	2	DH-BE-0020	1,7			DH-ZA-P010	303,6			DH-SC-0030	607,2
PVO19003038	15	3	DH-BE-0020	17,8							DH-SC-0030	4332,6

Order nr.	Due Day	Team Size	Task (Gathering)	m3	Task (Short)	m3	Task (Saw 1)	m1	Task (Saw 2)	m1	Task (Plan)	m1
PVO19003747	15	2	DH-BE-0010	3,4	DH-KO-0005	3,4	DH-ZA-P010	525,6			DH-SC-0030	1051,2
PVO19003874	15	2	DH-BE-0020	0,3			DH-ZA-S010	182,7			DH-SC-0030	365,4
PVO19003875	15	3	DH-BE-0010	1,9	DH-KO-0005	1,9					DH-SC-0030	332,8
PVO19003892	15	3	DH-BE-0010	3,6	DH-KO-0005	3,8					DH-SC-0030	946,0
PVO19003909	15	2	DH-BE-0020	1,2			DH-ZA-K010	459,9			DH-SC-0040	919,8
PVO19003952	15	3	DH-BE-0020	0,5			DH-ZA-K010	72,0			DH-SC-0030	144,0
PVO19003960	15	3	DH-BE-0010	1,9	DH-KO-0005	1,9	DH-ZA-K010	442,2			DH-SC-0030	884,4
PVO19003961	15	2	DH-BE-0020	3,8	DH-KO-0005	3,8	DH-ZA-S010	1493,7			DH-SC-0030	2987,4
PVO19003557	16	2	DH-BE-0020	6,2							DH-SC-0030	2252,3
PVO19003672	16	3	DH-BE-0020	5,8	DH-KO-0005	5,8	DH-ZA-S020	1342,3			DH-SC-0030	4026,9
PVO19003675	16	3	DH-BE-0020	1,3							DH-SC-0030	262,5
PVO19003677	16	2	DH-BE-0020	0,5			DH-ZA-S010	181,3			DH-SC-0030	362,6
PVO19003746	16	2	DH-BE-0020	0,5			DH-ZA-S010	239,3			DH-SC-0040	478,5
PVO19003748	16	3	DH-BE-0010	1,1	DH-KO-0005	1,1	DH-ZA-K010	272,4			DH-SC-0060	544,8
PVO19003749	16	2	DH-BE-0010	1,2							DH-SC-0040	615,6
PVO19003750	16	2	DH-BE-0020	2,8			DH-ZA-S010	706,7			DH-SC-0030	1413,3
PVO19003755	16	3	DH-BE-0010	1,0							DH-SC-0080	526,5
PVO19003923	16	2	DH-BE-0020	9,7							DH-SC-0030	5622,0
PVO19003548	17	2	DH-BE-0020	3,2			DH-ZA-S010	1258,1			DH-SC-0030	2516,2
PVO19003549	17	2	DH-BE-0020	0,7			DH-ZA-S010	349,6			DH-SC-0040	699,2
PVO19003598	17	2	DH-BE-0020	3,1			DH-ZA-S010	1216,0			DH-SC-0030	2432,0
PVO19003692	17	3	DH-BE-0020	0,4			DH-ZA-S020	160,9			DH-SC-0030	482,7
PVO19003757	17	3	DH-BE-0010	1,0			DH-ZA-K020	244,0			DH-SC-0040	732,0
PVO19003758	17	2	DH-BE-0020	0,3			DH-ZA-S010	183,0			DH-SC-0030	366,0
PVO19003759	17	3	DH-BE-0020	0,6			DH-ZA-S020	167,3			DH-SC-0030	501,9
PVO19003760	17	3	DH-BE-0020	3,9			DH-ZA-K020	479,0	DH-ZA-P010	1437,0	DH-SC-0040	2874,0
PVO19003795	17	2	DH-BE-0010	4,9	DH-KO-0005	4,9	DH-ZA-P010	891,0			DH-SC-0030	1782,0
PVO19003796	17	2	DH-BE-0010	1,2			DH-ZA-P010	462,0			DH-SC-0030	924,0
PVO19003813	17	3	DH-BE-0010	2,4	DH-KO-0005	2,4	DH-ZA-K010	683,3			DH-SC-0030	1366,6
PVO19003814	17	3	DH-BE-0020	1,3	DH-KO-0005	1,3	DH-ZA-S010	493,5			DH-SC-0050	987,0
PVO19003817	17	2	DH-BE-0020	3,2			DH-ZA-S010	1193,8			DH-SC-0030	2387,6
PVO19003818	17	3	DH-BE-0020	1,9	DH-KO-0005	1,9	DH-ZA-S010	709,5			DH-SC-0050	1419,0
PVO19003889	17	2	DH-BE-0020	0,3			DH-ZA-S010	244,0			DH-SC-0030	488,0
PVO19003890	17	2	DH-BE-0020	0,3			DH-ZA-S010	122,4			DH-SC-0040	244,7

Order nr.	Due Day	Team Size	Task (Gathering)	m3	Task (Short)	m3	Task (Saw 1)	m1	Task (Saw 2)	m1	Task (Plan)	m1
PVO19003894	17	2	DH-BE-0020	0,1			DH-ZA-S010	58,0			DH-SC-0040	115,9
PVO19003895	17	2	DH-BE-0020	1,0			DH-ZA-S010	609,4			DH-SC-0040	1218,7
PVO19003900	17	2	DH-BE-0020	0,3			DH-ZA-K010	73,5	DH-ZA-S010	147,0	DH-SC-0030	294,0
PVO19003427	18	3	DH-BE-0020	3,2			DH-ZA-K010	429,5	DH-ZA-S020	859,0	DH-SC-0040	2577,0
PVO19003596	18	3	DH-BE-0020	1,2							DH-SC-0030	241,2
PVO19003794	18	2	DH-BE-0020	2,8			DH-ZA-S010	1392,4			DH-SC-0030	2784,8
PVO19003873	18	3	DH-BE-0020	2,2			DH-ZA-K010	517,5			DH-SC-0050	1035,0
PVO19003876	18	2	DH-BE-0020	0,4			DH-ZA-S010	156,0			DH-SC-0030	312,0
PVO19003877	18	3	DH-BE-0010	0,4			DH-ZA-K010	197,1			DH-SC-0050	394,2
PVO19003879	18	3	DH-BE-0020	0,3			DH-ZA-S030	122,5			DH-SC-0030	490,0
PVO19003903	18	2	DH-BE-0010	1,3			DH-ZA-K010	675,0			DH-SC-0040	1350,0
PVO19003904	18	2	DH-BE-0020	12,4	DH-KO-0005	12,4	DH-ZA-P010	2644,5				
PVO19003968	18	3	DH-BE-0010	0,5							DH-SC-0060	189,1
PVO19003973	18	3	DH-BE-0010	0,3							DH-SC-0050	100,7
PVO19003602	19	3	DH-BE-0020	27,7			DH-ZA-K020	1286,3				
PVO19003643	19	3	DH-BE-0010	1,0			DH-ZA-K010	502,2			DH-SC-0050	1004,4
PVO19003667	19	2	DH-BE-0020	0,9							DH-SC-0040	690,8
PVO19003668	19	2	DH-BE-0020	1,3			DH-ZA-S010	681,6			DH-SC-0040	1363,2
PVO19003688	19	2	DH-BE-0020	1,7							DH-SC-0030	996,0
PVO19003756	19	3	DH-BE-0020	0,6			DH-ZA-P030	72,0			DH-SC-0040	288,0
PVO19003862	19	2	DH-BE-0020	3,7			DH-ZA-S010	1191,2			DH-SC-0040	2382,4
PVO19003878	19	3	DH-BE-0020	0,2							DH-SC-0030	55,5
PVO19003882	19	2	DH-BE-0020	1,7			DH-ZA-S010	828,3			DH-SC-0040	1656,6
PVO19003962	20	3	DH-BE-0020	2,8			DH-ZA-S020	993,6			DH-SC-0040	2980,8
PVO19003980	20	2	DH-BE-0020	0,6			DH-ZA-S010	329,3			DH-SC-0040	658,6
PVO19003981	20	3	DH-BE-0020	3,9			DH-ZA-K010	595,9	DH-ZA-P020	1191,8	DH-SC-0030	3575,4
PVO19003988	20	2	DH-BE-0020	4,9			DH-ZA-K010	1173,5			DH-SC-0030	2347,0
PVO19003995	20	2	DH-BE-0020	0,5			DH-ZA-S010	165,0			DH-SC-0030	330,0
PVO19003996	20	2	DH-BE-0020	0,9							DH-SC-0040	477,6
PVO19003999	20	3	DH-BE-0020	0,5			DH-ZA-K010	64,9	DH-ZA-P020	129,8	DH-SC-0040	389,4
PVO19004002	20	2	DH-BE-0020	1,2			DH-ZA-K010	234,9			DH-SC-0010	469,7
PVO19004004	20	2	DH-BE-0020	1,4			DH-ZA-S010	390,4			DH-SC-0030	780,8

Appendix D - Simulation Output

Order nr.	Wood	Due day	Task (Gathrting)	m3	Start Time	End Time	Task (Short)	m3	Start Time	End Time	Task(Saw)	m1	Start Time	End Time	Task(Saw)	m1	Start Time	End Time	Task (Plan)	m1	Start Time	End Time	Start Day	Start Time	End Day	End Time
PVO19003350	Accoya	1	DH-BE-0020	0,8	07:30:00	07:41:00									DH-ZA-S030	148,8	07:30:00	08:05:00	DH-SC-0040	595,2	07:49:00	09:03:00	1	07:30:00	1	09:03:00
PVO19003435	Cambara	1	DH-BE-0020	2,7	07:30:00	08:10:00									DH-ZA-S010	427,7	07:54:00	08:50:00	DH-SC-0040	855,4	08:34:00	10:06:00	1	07:30:00	1	10:06:00
PVO19003248	DRM	1	DH-BE-0010	3,1	07:30:00	08:16:00													DH-SC-0030	1171,2	08:00:00	09:40:00	1	07:30:00	1	09:40:00
PVO19003414	Vuren	1	DH-BE-0010	1,4	07:30:00	07:52:00	DH-KO-0005	1,4	07:47:00	08:17:00					DH-ZA-S010	198	08:01:00	08:44:00	DH-SC-0050	396	08:28:00	09:27:00	1	07:30:00	1	09:27:00
PVO19003625	DRM	1	DH-BE-0020	0,2	08:48:00	09:18:00					DH-ZA-K010	36	09:02:00	09:36:00	DH-ZA-S020	72	09:20:00	09:55:00	DH-SC-0030	216	09:39:00	10:23:00	1	08:48:00	1	10:23:00
PVO19003674	DRM	1	DH-BE-0010	1,9	09:12:00	09:42:00	DH-KO-0005	1,9	09:37:00	10:07:00									DH-SC-0030	332,75	09:51:00	10:42:00	1	09:12:00	1	10:42:00
PVO19003814	DRM	1	DH-BE-0020	12,4	09:25:00	12:31:00	DH-KO-0005	12,4	12:26:00	13:45:00	DH-ZA-P010	2644,5	13:29:00	09:09:00									1	09:25:00	2	09:09:00
PVO19003405	DRM	2	DH-BE-0010	0,9	09:51:00	10:21:00	DH-KO-0005	0,9	10:16:00	10:46:00									DH-SC-0030	266,6	10:30:00	11:17:00	1	09:51:00	1	11:17:00
PVO19003510	DRM	2	DH-BE-0020	1,6	10:08:00	10:38:00													DH-SC-0010	630,3	10:26:00	11:28:00	1	10:08:00	1	12:20:00
PVO19003464	Accoya	2	DH-BE-0020	1,0	10:27:00	10:57:00									DH-ZA-S010	319,2	10:41:00	11:31:00	DH-SC-0030	638,4	11:15:00	12:24:00	1	10:27:00	1	12:24:00
PVO19003671	DRM	2	DH-BE-0010	1,9	11:02:00	11:32:00	DH-KO-0005	1,9	11:27:00	11:57:00	DH-ZA-K010	442,2	11:41:00	12:44:00					DH-SC-0030	884,4	12:28:00	13:51:00	1	11:02:00	1	13:51:00
PVO19003875	DRM	2	DH-BE-0020	5,8	12:05:00	13:31:00	DH-KO-0005	5,8	13:45:00	14:25:00					DH-ZA-S020	1342,3	14:09:00	07:59:00	DH-SC-0030	4026,9	07:43:00	12:11:00	1	12:05:00	2	12:11:00
PVO19003404	DRM	3	DH-BE-0010	1,1	12:09:00	12:39:00	DH-KO-0005	1,1	14:25:00	14:55:00									DH-SC-0030	363,35	14:39:00	07:31:00	1	12:09:00	2	07:31:00
PVO19003484	DRM	3	DH-BE-0020	2,9	13:36:00	14:19:00									DH-ZA-S010	1495	14:03:00	08:02:00	DH-SC-0040	2990	07:46:00	11:51:00	1	13:36:00	2	11:51:00

Figure .1: Output Window of a Simulation Run

Appendix E - Detailed Results Scenario 1

Table .4: Detailed results of QRM-Based EDD scenario (part 1)

	Fixed-3				Fixed-2/3				Variable-2/3			
nr.	**Makesp.**	**Delay**	**# del.**	**Wait**	**Makesp.**	**Delay**	**# del.**	**Wait**	**Makesp.**	**Delay**	**# del.**	**Wait**
1	154.8	102.3	48	2.9	144.9	24.7	8	3.7	145.9	79.0	26	4.6
2	155.7	159.9	58	5.3	144.5	71.0	30	6.5	150.4	269.9	51	5.2
3	152.9	289.9	87	1.5	143.6	176.0	30	8.7	141.7	160.5	27	3.0
4	154.7	253.4	69	2.0	143.7	66.0	25	4.7	148.7	162.3	42	4.6
5	154.8	60.3	22	5.1	143.1	13.3	7	2.5	150.9	10.1	6	2.7
6	155.1	118.0	41	1.3	144.8	97.7	22	5.3	151.9	98.7	24	2.1
7	156.0	53.2	21	5.7	145.2	0.6	1	7.5	149.3	36.5	11	5.1
8	152.2	323.9	104	1.3	142.1	149.6	40	4.9	149.3	36.5	11	5.1
9	153.7	203.6	71	5.8	141.9	79.0	26	3.7	149.9	197.8	55	3.7
10	153.7	243.4	91	1.3	143.9	85.0	38	7.3	148.9	120.3	44	5.6
11	154.4	273.8	82	4.6	143.2	149.6	37	6.1	143.0	149.6	37	5.8
12	152.9	377.7	109	2.9	143.8	91.8	35	5.8	147.8	168.0	42	3.8
13	156.2	270.9	80	7.5	144.0	274.7	64	3.0	144.0	272.7	64	3.0
14	155.3	580.8	133	0.5	145.0	314.9	67	5.7	150.9	325.4	82	4.6
15	153.9	92.1	39	1.5	143.1	84.2	25	4.0	155.2	44.9	14	7.6
16	154.8	514.9	113	3.4	144.1	131.0	44	4.5	148.5	188.0	58	6.5
17	156.0	17.6	6	2.5	145.5	1.7	1	7.2	149.2	79.0	4	2.3
18	154.9	303.8	92	4.1	142.7	55.7	25	2.1	149.4	388.1	74	2.1
19	152.8	238.5	86	2.4	142.9	40.1	19	6.5	149.6	346.7	63	5.7
20	156.0	27.5	13	5.7	142.8	0.0	0	4.1	154.7	11.7	5	2.5
21	153.1	498.1	129	4.2	142.5	253.8	63	4.2	154.7	211.3	54	4.3
22	153.8	615.5	114	4.9	142.5	59.2	22	3.5	151.3	459.9	68	4.3
23	153.0	315.4	93	0.8	143.8	217.8	55	6.6	156.4	214.0	58	2.9
24	154.2	69.7	34	5.0	145.8	105.6	24	12.2	153.5	116.2	29	8.2
25	152.3	115.4	46	2.2	142.2	100.6	28	6.1	145.0	172.3	41	4.5
26	153.8	110.0	51	2.6	144.6	42.0	18	3.7	144.4	56.8	26	4.0
27	153.9	56.5	21	3.1	143.7	64.8	25	5.5	142.1	121.8	37	4.4
28	157.4	122.3	49	4.3	145.1	77.1	30	6.7	150.8	164.1	43	6.1
29	155.1	251.3	79	3.4	143.2	98.7	37	4.7	150.2	182.5	58	4.9
30	152.8	336.7	90	4.3	143.0	64.5	27	7.8	146.5	166.8	45	11.7
31	156.5	551.7	114	5.0	145.7	114.0	48	5.3	145.2	138.7	50	7.4
32	152.8	125.1	52	2.3	145.1	98.6	28	8.7	150.8	112.7	33	4.2
33	152.8	731.8	126	4.0	145.8	321.4	73	12.8	146.1	347.3	74	10.4
34	153.2	65.5	28	1.1	143.3	51.5	19	7.2	154.7	78.1	27	2.3
35	154.3	157.3	59	4.8	142.4	20.1	8	5.4	142.4	44.1	20	4.1

Table .5: Detailed results of QRM-Based EDD scenario (part 2)

	Fixed-3				**Fixed-2/3**				**Variable-2/3**			
nr.	**Makesp.**	**Delay**	**# del.**	**Wait**	**Makesp.**	**Delay**	**# del.**	**Wait**	**Makesp.**	**Delay**	**# del.**	**Wait**
36	152.8	432.1	96	3.0	143.2	227.1	51	6.7	145.3	328.4	59	3.9
37	156.8	708.8	134	6.3	143.5	366.9	57	4.9	147.8	576.3	67	4.2
38	154.7	46.4	23	2.2	144.9	7.1	7	7.7	149.5	40.4	20	6.2
39	153.3	140.8	62	2.0	144.7	62.7	25	5.3	150.4	66.6	31	4.4
40	153.9	185.3	61	3.7	143.2	301.5	56	2.1	159.4	284.8	45	3.8
41	154.5	181.2	64	2.3	146.0	121.6	33	6.9	148.0	243.2	56	6.5
42	154.1	304.1	99	3.0	145.1	89.2	35	6.7	153.4	227.5	51	6.4
43	154.5	154.8	56	4.1	142.7	26.5	14	3.6	148.9	47.6	19	9.5
44	154.2	142.9	59	2.0	143.8	29.4	14	5.7	144.6	101.6	41	2.7
45	153.4	438.7	100	3.4	142.4	258.8	54	3.2	150.2	286.0	67	6.2
46	153.8	367.1	83	3.4	143.3	151.8	46	3.7	153.7	196.4	47	5.7
47	153.4	330.7	94	4.7	142.7	54.4	24	7.4	147.5	93.3	34	6.7
48	153.3	397.2	122	3.7	144.6	372.4	63	5.6	157.3	369.5	84	3.7
49	151.4	109.2	55	0.2	141.9	64.2	30	3.4	152.8	128.8	46	4.0
50	155.6	343.6	96	1.6	144.9	191.3	47	4.5	144.8	348.2	64	6.8
51	153.9	84.2	35	6.2	142.2	85.2	26	4.3	156.3	92.0	30	5.4
52	155.4	421.4	112	3.6	145.8	147.1	48	4.8	149.7	125.3	38	2.3
53	153.9	138.9	46	1.5	143.9	89.6	25	3.7	147.3	203.9	40	4.5
54	152.6	155.9	57	0.7	145.0	120.7	30	5.3	156.4	100.7	34	5.0
55	155.7	517.1	124	1.9	144.0	232.2	64	5.8	152.3	155.4	53	2.4
56	154.4	527.6	122	5.8	141.8	114.6	31	4.8	146.0	139.9	37	4.4
57	152.3	219.2	80	2.4	142.7	50.0	27	4.2	149.6	362.4	68	5.2
58	152.8	35.0	18	3.4	145.1	38.5	15	5.5	149.5	8.9	7	1.6
59	154.0	383.5	101	6.9	143.3	115.1	36	7.5	148.3	195.2	45	3.2
60	155.7	55.3	27	3.8	143.7	72.8	19	4.8	143.7	72.8	19	4.8

Appendix F - Detailed Results Scenario 2

Table .6: Detailed results of QRM-Based EDD-SPT scenario (part 1)

	Fixed-3				**Fixed-2/3**			
nr.	**Makesp.**	**Delay**	**# del.**	**Wait**	**Makesp.**	**Delay**	**# del.**	**Wait**
1	154.7	80.9	31	0.3	143.4	25.1	8	1.8
2	154.3	142.1	43	4.9	144.2	51.7	18	4.6
3	152.7	209.1	53	0.9	142.6	134.6	29	4.0
4	154.0	202.4	49	1.4	143.8	44.2	20	2.3
5	155.3	53.3	21	2.1	145.8	21.1	9	6.1
6	156.3	93.5	33	1.5	145.9	65.8	20	4.4
7	155.2	45.9	17	2.1	146.0	3.9	4	5.4
8	158.3	307.9	77	4.7	148.5	130.6	31	5.2
9	156.3	147.6	44	2.6	149.1	73.9	24	8.2
10	156.4	211.3	64	4.1	145.5	66.4	22	3.7
11	154.7	179.5	55	2.5	142.9	132.2	31	4.0
12	153.3	272.3	77	3.3	143.6	82.3	28	6.4
13	157.1	180.8	57	3.5	147.5	271.7	56	7.6
14	157.4	508.7	111	4.4	144.4	229.7	55	6.8
15	154.7	89.5	33	2.1	143.9	72.0	20	7.1
16	156.0	418.0	90	3.6	145.3	144.4	40	5.9
17	156.7	15.5	5	2.4	145.8	1.6	1	5.7
18	153.5	214.1	63	1.0	143.6	65.3	23	3.3
19	154.8	259.2	71	5.0	143.7	49.8	17	3.0
20	156.4	16.3	7	0.9	146.4	5.7	5	10.1
21	155.5	347.3	87	2.5	142.9	206.9	46	3.0
22	153.8	422.8	90	2.6	142.3	74.2	28	3.8
23	153.4	247.3	65	2.1	148.2	143.2	45	3.6
24	154.8	63.2	23	3.4	145.2	67.2	14	5.5
25	152.8	91.2	29	0.8	143.5	84.7	24	2.5
26	155.2	110.6	41	2.9	145.3	51.5	18	4.2
27	153.9	68.6	22	2.0	144.3	95.4	31	7.4
28	155.2	103.6	36	1.9	146.1	79.5	21	8.2
29	155.9	200.2	56	1.7	145.6	73.7	22	1.9
30	153.0	265.3	72	3.4	141.9	50.3	20	4.4
31	154.9	385.0	85	2.4	145.4	109.2	31	8.4
32	154.7	124.6	39	3.5	144.3	71.3	19	5.5
33	154.3	545.9	96	3.0	142.6	149.4	44	4.2
34	155.1	76.7	24	2.7	143.3	52.5	12	4.1
35	156.2	171.1	52	6.9	144.9	20.4	10	4.9

Table .7: Detailed results of QRM-Based EDD-SPT scenario (part 2)

	Fixed-3				**Fixed-2/3**			
nr.	**Makesp.**	**Delay**	**# del.**	**Wait**	**Makesp.**	**Delay**	**# del.**	**Wait**
36	155.8	363.5	80	1.8	143.8	214.9	44	4.2
37	156.9	523.7	106	3.9	146.1	286.6	46	5.2
38	155.1	53.5	24	1.3	144.7	16.8	9	6.6
39	152.9	117.9	44	0.9	142.5	55.6	20	4.0
40	155.5	152.9	49	3.0	144.8	246.3	48	4.5
41	157.5	176.7	53	3.2	149.1	89.8	31	7.9
42	153.0	202.3	56	1.3	143.0	77.2	28	2.9
43	153.4	127.4	36	2.9	142.8	31.9	8	4.4
44	154.3	153.2	50	3.4	144.1	57.6	20	8.1
45	154.3	339.2	75	1.7	144.3	210.3	44	5.8
46	153.5	271.5	60	1.7	143.4	154.7	36	4.6
47	156.5	249.0	66	3.9	143.6	76.7	22	6.8
48	152.9	253.9	67	1.2	143.1	291.6	48	5.4
49	153.9	122.6	41	2.0	142.7	62.3	21	1.7
50	155.0	272.6	72	2.1	144.4	139.2	33	2.5
51	156.1	53.1	24	1.8	143.5	80.2	21	5.5
52	158.0	344.9	83	3.1	147.0	101.7	30	1.7
53	154.7	105.2	33	0.7	142.4	61.3	19	3.1
54	153.9	156.6	45	2.3	143.1	103.5	24	2.5
55	154.9	421.6	91	1.6	142.9	214.7	56	6.0
56	153.2	354.4	84	3.5	142.8	96.8	27	3.2
57	153.3	187.1	59	1.9	143.8	52.5	22	7.5
58	155.2	37.5	14	2.0	144.2	38.6	10	4.2
59	156.7	241.4	60	2.1	146.9	89.7	24	6.9
60	155.8	52.2	18	4.1	142.7	53.5	17	1.6

Appendix G - Detailed Results Scenario 3

Table .8: Detailed results of QRM-Based GA scenario

	Fixed-3		Fixed-2/3			Fixed-3		Fixed-2/3	
nr.	**Makesp.**	**Delay**	**Makesp.**	**Delay**	**nr.**	**Makesp.**	**Delay**	**Makesp.**	**Delay**
1	154.3	75.6	142.3	38.1	31	154.3	362.0	145.3	91.9
2	152.6	102.0	143.9	36.5	32	153.7	98.2	143.2	77.7
3	151.8	197.8	141.5	124.1	33	153.2	489.2	142.3	129.0
4	153.5	183.5	143.2	41.6	34	152.4	65.9	141.0	46.0
5	154.7	43.5	141.8	13.0	35	154.2	116.5	144.7	15.7
6	154.0	87.3	143.3	63.7	36	155.2	328.7	143.2	185.6
7	154.1	35.3	142.9	2.6	37	156.1	362.6	145.1	204.3
8	157.0	268.5	146.2	121.5	38	153.5	36.7	144.1	11.5
9	155.7	135.1	145.7	59.8	39	151.6	112.6	141.6	53.4
10	154.4	189.4	142.9	63.9	40	152.8	134.9	142.4	226.2
11	153.9	158.8	141.3	96.5	41	156.6	154.0	147.4	70.6
12	151.7	253.0	142.0	72.1	42	152.8	201.4	142.7	72.4
13	155.9	154.2	146.4	189.8	43	152.3	99.5	141.4	25.0
14	154.3	406.1	143.0	223.1	44	153.3	121.5	143.1	35.2
15	153.3	76.0	142.8	64.1	45	152.9	275.3	142.9	196.5
16	155.3	362.7	145.0	135.2	46	153.1	238.8	141.9	144.9
17	155.1	11.0	142.8	1.0	47	155.3	196.1	142.3	56.7
18	152.9	202.5	143.5	49.1	48	152.0	525.6	142.3	245.2
19	153.1	183.1	141.7	37.9	49	153.7	114.7	142.2	50.2
20	153.3	12.5	142.4	1.1	50	153.2	254.6	141.2	124.4
21	155.4	306.6	141.6	193.3	51	155.5	43.4	141.2	63.2
22	152.0	384.6	142.1	50.3	52	157.2	306.0	146.9	97.6
23	152.7	229.5	142.8	172.3	53	154.3	96.8	142.0	57.8
24	153.4	51.4	143.7	61.4	54	152.6	145.8	142.7	93.3
25	152.5	82.0	144.7	62.9	55	153.2	363.4	141.6	180.4
26	152.9	94.3	143.5	43.3	56	151.8	293.0	142.4	89.1
27	153.1	61.4	143.0	65.1	57	153.0	170.6	143.5	43.2
28	154.4	92.4	142.8	64.4	58	155.1	36.2	141.8	36.0
29	155.3	175.2	144.8	71.7	59	155.3	209.6	146.4	79.9
30	152.2	283.3	140.8	41.3	60	152.5	41.8	142.4	52.2

Appendix H - Detailed Results Scenario 4

Table .9: Results for the total makespan (in hours) for Two-Stage SPT-EDD scenario

Team	Buffer	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-2/3	$b = 6$	141.6	159.2	4.0	149.1	148.1	150.1
	$b = 8$	141.8	156.6	3.9	148.2	147.2	149.2
	$b = 10$	141.7	159.5	4.1	148.7	147.6	149.7
	$b = 12$	141.5	158.1	3.9	148.6	147.6	149.6
	$b = 14$	140.3	156.8	3.9	148.4	147.4	149.4
	$b = 16$	141.1	163.7	4.7	149.1	148.0	150.3
	$b = 18$	140.5	159.4	4.4	149.2	148.1	150.3
	$b = 20$	140.6	160.0	4.4	149.6	148.5	150.7
Variable-2/3	$b = 6$	142.1	154.6	2.6	146.4	145.8	147.1
	$b = 8$	142.5	155.4	2.7	146.0	145.4	146.7
	$b = 10$	141.6	154.9	2.6	145.8	145.2	146.5
	$b = 12$	141.7	150.5	2.0	145.5	145.0	146.0
	$b = 14$	142.0	151.0	2.2	145.5	144.9	146.0
	$b = 16$	141.3	151.0	2.2	145.2	144.6	145.7
	$b = 18$	141.4	153.0	2.4	145.0	144.4	145.6
	$b = 20$	141.4	151.4	2.2	145.2	144.6	145.7

Table .10: Results for the total delay (in hours) for Two-Stage SPT-EDD scenario

Team	Buffer	Min.	Max.	Std. Dev.	Avg.	Lower bound	Upper bound
Fixed-2/3	$b = 6$	1.3	303.7	69.0	86.9	69.5	104.4
	$b = 8$	2.0	345.2	69.8	88.0	70.4	105.7
	$b = 10$	1.5	256.1	65.7	91.5	74.8	108.1
	$b = 12$	1.7	336.4	73.3	96.7	78.1	115.2
	$b = 14$	3.2	309.2	73.9	101.9	83.2	120.6
	$b = 16$	4.5	397.2	79.8	109.3	89.1	129.5
	$b = 18$	4.2	427.3	89.7	121.1	98.4	143.8
	$b = 20$	4.8	433.8	92.4	129.3	105.9	152.7
Variable-2/3	$b = 6$	0.4	207.1	47.4	56.8	44.8	68.8
	$b = 8$	0.0	203.7	48.1	56.2	44.0	68.4
	$b = 10$	1.4	217.1	47.0	54.4	42.5	66.3
	$b = 12$	0.0	193.3	45.3	52.0	40.5	63.5
	$b = 14$	0.0	181.4	43.1	50.5	39.6	61.4
	$b = 16$	0.1	166.8	41.8	48.5	37.9	59.1
	$b = 18$	0.6	166.7	42.0	47.9	37.3	58.5
	$b = 20$	0.6	186.3	44.0	48.4	37.3	59.5