MASTER

# A two-step Order Acceptance and Scheduling Approach using Branch-and-Price for make-to-order Manufacturing Systems

Ruiter, S.J.K.

*Award date:*
2021

Department Industrial Engineering & Innovation Sciences
Information Systems Research Group
Operations management and logistics

# A two-step Order Acceptance and Scheduling Approach using Branch-and-Price for make-to-order Manufacturing Systems

*Master Thesis*

S.J.K. Ruiter

| | |
|---|---|
| First supervisor: | Dr. Murat Firat |
| Second Supervisor: | Dr. Ahmadreza Marandi |
| Company Supervisor: | Ir. Dirk van Driel |

Student Number: 1376012

Version 2

Eindhoven, September 2021

Keywords: Flexible Job Shop Scheduling, Order Acceptance and Scheduling, Make-To-Order, Workload Optimization, Branch-and-Price

# Abstract

This report presents a two-step approach to the Order Acceptance and Scheduling (OAS) problem of a make-to-order manufacturing system. The considered manufacturing system has flexible job shop characteristics with complicated processing routes of final products. The diversity in product types, hence customers, is an advantage for a make-to-order production system in being robust to fluctuations in market conditions. However, this brings a challenge in planning the operations so that different types of products get produced simultaneously and efficiently in a given time horizon. This project is mainly about developing an automated planning system to tackle the aforementioned challenge.

Since the company under consideration has stable business relations with customers and the model finds short-term schedules, it is assumed that the problem set has deterministic data. All the operations in the production system are assumed to be deterministic as well. Last-minute order changes from customers are neglected, e.g. cancellations and amount changes. The dynamic aspect of the application is left partially to business implementation and to further studies under the collaboration of the company with its academic partners.

The proposed OAS approach is performed in two steps. The first-level workload optimization approach consists of a Mixed-Integer Linear Program (MILP) to determine orders accept and the required deadline through setting system-aware production targets. The Second-level scheduling approach is realized by using a branch-and-price framework which solves the flexible job shop scheduling problem. The model determines how to schedule the orders in order to minimize the deadline violation.

Computational analyses show that the proposed OAS model is well able to determine the production target of orders to be produced, and based on the production target, create a production schedule of the corresponding orders. The model is benchmarked based on the predicted on-time delivery of orders. The results show the desired on-time delivery targets can be achieved using the proposed two-step OAS approach.

# Executive summary

In this master thesis, an order acceptance and scheduling model is developed for a make-to-order manufacturing company where production has flexible job shop characteristics with the goal of minimizing order tardiness. The OAS model is implemented as a case study at AME, an independent developer, and manufacturer of high-quality electronic products. With AME's current planning method, on average 76% of the orders are delivered on time. However, AME aims for a target of 85%. The goal of the OAS approach is to improve the production performance so the target can be achieved.

An order is a customer's request for a deterministic quantity of a final product. This product consists of a composition of sub-products, and these sub-products in turn consist of a composition of sub-products. This relationship can be seen as a precedence relationship since the final product cannot be produced until the sub-products are produced. A product can be produced by performing a set of operations. For each operation, a set of machines on which the operation can be performed is defined. A job here represents the execution of the required operation for the desired order quantity. When the production of a job is started it cannot be interrupted prematurely, preemption is not allowed. Orders have deterministic order sizes and order deadlines, each order consists of a set of jobs. Each job has a deterministic processing time and the setup time is included in the processing time.

The scheduling model considers a set of schedulable orders. The goal when realizing the planning is to minimize the tardiness of the orders. First, the model ensures that if the machine capacities are overloaded, orders are rejected prematurely so the maximum utilization is not exceeded. Secondly, the accepted orders must be scheduled in the production schedule.

The proposed OAS approach consists of a two-level production planning (Figure 1). In the first level, the workload optimization model makes order acceptance decisions and determines the production targets with precise deadlines of sub-components to be produced for accepted orders. The scheduling model of the second level finds schedules of operations by employing decomposition techniques from mathematical optimization to minimize the tardiness of the production targets.

The workload optimization problem is formulated as Mixed Integer Linear Programming (MILP) model. The model respects the capacity of machines and determines target deliveries of accepted orders, possibly by delaying the original deadlines within the customer tolerance. If the deadline of the order is still infeasible, the model can choose to reject the order. The results show the workload control of resources is accomplished by rejecting orders. Based on the results, it can be concluded that the model is able to achieve the workload control.

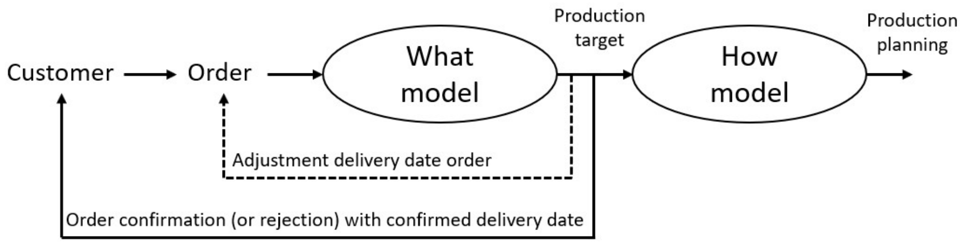



Figure 1: Overview two-level production planning

The second step scheduling model includes a branch-and-price framework. Branch-and-price uses Column Generation for finding lower bounds at nodes of the branch-and-bound search tree. The scheduling problem is formulated as a master problem that is solved to optimality via a branch-and-price search. The objective of the master model is to minimize the sum of tardiness of orders. The master problem chooses a schedule for each order from the set of alternative schedules of the order. By solving the master problem, the duals are generated. When the pricing problem is solved using these duals as input, a new schedule is generated for each order. If these are found to have a negative reduced cost, this schedule may have the potential to improve the objective of the master problem and is added to the set of possible schedules of an order. This process keeps repeating until no orders are found with negative reduced costs. If no promising columns are found, the optimal solution is found and the node explored. However, due to the relaxation of the variables, the model may still consist of fractional values. This creates an unfeasible solution. By applying a branch-and-Bound optimization method, the search space can then be explored to reach an integer solution. For the branching, the highest threshold selection method is applied to define the schedule to be fixed, and the search tree is explored using a dept-first search method.

Based on the results of the implemented framework, it can be concluded that the proposed branch-and-price framework is well able to minimize the tardiness of orders. There is a large decrease in the objective tardiness of the greedy planning method and the branch-and-price framework. The model performs as intended, the on-time delivery performance is improved and the percentage of on-time completion of orders increased compared to the current scheduling method.

Based on an analysis of the previous planning method, an average of 76% of orders was delivered on time. However, AME aims to deliver 85% of the orders on time. The computational results have been analyzed and, based on these results, it is found that by implementing the two-step OAS approach, on-time delivery of 92% can be achieved. By implementing the OAS model the targeted on-time delivery performance is achieved.

A further recommendation for AME is to enhance the precision of job execution cycle timings. AME has a good collection of process time, however, the precision of the execution data might be enhanced. The Manufacturing Execution System will be especially useful for automatically recording job executions so that the planning models can leverage dynamically changing cycle times.

# Preface

I am proud to present my master thesis. The submission of my master thesis means that I can proudly say that I have completed the final requirements for my master's degree in Operations Management and Logistics with a specialization in Information Systems at the Eindhoven University of Technology. I am very proud of my work, however, I could not have accomplished this without the support of other people. Therefore I would like to take this opportunity to thank the people who helped and supported me during this last phase of my studies. I certainly did not get away with it easily and worked hard to achieve my goal.

Murat Firat, first of all, I would like to thank you for our cooperation. It wasn't always easy but you helped me through this very well. I am very happy with the cooperation and the feedback you have given me. During the master, our cooperation actually already started when I started as a tutor for one of your courses. I am very happy that I chose you as my mentor at that time!

Furthermore, I would like to thank AME, and Dirk van Driel in particular for making this thesis possible. During my thesis, I was able to consult with Erinc Albey and Ahmet Sahin, for which I would also like to express my gratitude.

At last, I would also like to thank my friends and family. Too bad nobody understood what I was doing but nevertheless, you have always supported me, I will always be grateful to you.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Production scheduling has been a popular research topic in operation research. Production scheduling is an essential task to help companies to improve their production process. Companies that implement this automated scheduling approach can consequently perform further optimization through sensitivity analysis, this makes research in this area highly relevant for companies.

Companies are producing more and more customized products. This leads to an industry with high product diversity. Production with a high degree of interdependence between separate products results in working with a high diversity of different production paths. Creating an optimal planning has therefore become more complex. This problem is also known as the flexible manufacturing system (FMS) scheduling problem [2]. Scheduling problems are highly difficult to solve optimally and are found to be NP-hard [3].

Production scheduling in a complex production facility with order deadlines and capacitated production units. Due to the high diversity of production of orders and order requirements, it becomes challenging to define the planning considering the available capacity and required delivery deadlines. This challenge could be overcome via a prescriptive analysis to determine the optimal schedule.

Companies that implement an automated scheduling approach can consequently perform further optimization through sensitivity analysis. Therefore, it makes it very appealing for companies to consider an automatic scheduling approach.

The research is conducted as a case study at Applied Micro Electronics (AME). AME is an independent developer and manufacturer of high-quality electronic products located in the top technological region of the world (Brainport Eindhoven). AME's goal is to create innovative products that exceed customer expectations. AME does this by integrating product development and manufacturing and keeping a clear focus on the product and its function. Driven by technology, AME strives for the best solution combining the disciplines of electrical, mechanical, software, and industrial engineering. Through creativity, passion, ambition, motivation, and a highly educated level of their employees, AME secures its goal of being a profitable company [4].

The objective of this research is to create an Order Acceptance and Scheduling approach for the production process of AME by developing an optimization branch-and-price framework for the production scheduling that aims to optimize due-date-related objectives in order to minimize deadline violations of orders.

## 1.1 Automated planning approach at AME

The proposed production planning consists of a two-level production planning. The first-level planning is an approach to determines which orders to accept and the required deadline. If deadlines determined by the customer are infeasible, the deadline can be adjusted by the model. In the case of two many orders resulting in not enough machine capacity, the orders will be rejected. This first-level model is further referred to as the what-model because it determines what orders to accept. The resulting output of the what-model determines which orders have to be produced, which orders are canceled, and the deadline per work center. The output of the what-model will be further referred to as the production target. The Second-level model focuses on how to schedule the projects in order to minimize the deadline violation of the production target. The model is further referred to as the how-model. This study is a follow-up study to the work of Firat and de Meyere [5]. The previous research proposed a production target model. The proposed model will be expanded and used as a production target model. In addition, the focus of this research will be on the second-level scheduling model. The two-level production planning is shown schematically in Figure 4.1.

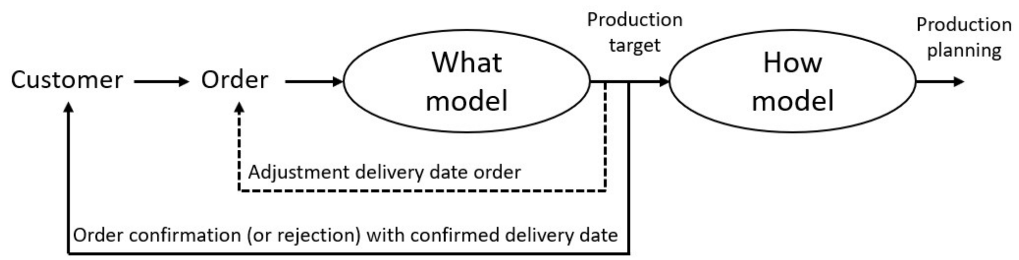



Figure 1.1: Overview two-level production planning

## 1.2 Research Objective

This research focuses on how to develop a two-step scheduling approach for an order acceptance and scheduling problem. The first model is used to determine the production target and define the accepted orders and their corresponding deadlines. The second model is to determine the schedule of the production target with the objective of minimizing the deadline violations of orders.

Based on the specifications of the different work centers, a planning must be constructed. This scheduling task is constrained by, for example, task deadlines, job sequences, machine capacity, and employee capacity. A schedule must be defined so that the AME can use this for planning the orders. Per job it should be specified when, and on which machine, the job should be performed. The production requires a robust approach of scheduling for the how-model.

For the first model to determine the production target, the model of Firat and Meyere [5] is used as a basis. In this study, the model is extended to improve the accuracy and efficiency of the model.

Because flexible job shop scheduling problems are computationally expensive to solve, the efficiency of the model must be considered. Branch-and-price is an effective approach to solving large-scale scheduling problems [1]. This study will focus on solving the how-model via the branch-and-price approach.

The goal of this research is to create an Order Acceptance and Scheduling approach for the production process of AME by developing an branch-and-price optimization framework for the production scheduling that aims to optimize due-date-related objectives in order to minimize deadline violations of orders.

The existing literature does not fully match the characteristics of the AME production process. Furthermore, the models in the literature are tested on smaller instance sizes and it is therefore questionable whether the approaches can handle the instance size of the AME case study.

This research will contribute to the current literature because there is currently no branch-and-price based method to generate an automatic scheduling approach for job shop type manufacturing companies that operate with make-to-order convention and have a high production diversity and characteristics as AME.

## 1.3 Research scope

The model to determine the production target proposed by Firat and Meyere will be used as what-model [5]. However, some adjustments are made to the model to provide a more realistic representation of the situation at AME. The generated production targets of the what-model form the problem instances of the planning model. The how-model is solved via the branch-and-price approach. The output of the model must be applicable for production. It must be clear to production in what order the products must be produced, an overview should be given of the number of products that must be produced, and when the products must be finished.

## 1.4 Research question

In this section, our project goal is clearly stated in the form of a research question and it is further decomposed into steps by formulating sub-questions. The main research question is formulated in the following way:

*How should the highly complex order acceptance and scheduling problem of AME be solved efficiently by using a branch-and-price search?*

In order to answer the main question with well-defined steps, several sub-questions are defined. The sub-questions are defined in the following way:

*1. What are the main characteristics of the production system of AME company and what is the current business practice for planning the manufacturing operations?*

First, the AME production process will be explained in more detail to get a better insight into the production process. Each work center will be explained with the requirements to achieve a realistic representation of the process. This question answered in Chapter 2

*2. What are the state-of-the-art approaches in flexible job shop scheduling in the literature and what are the comparable OAS approaches to the production situation of AME in literature?*

The second sub-question has two purposes. First, the current methods in the literature are examined for scheduling of production systems. This complements the main question because it clarifies which optimization techniques can be applied. The advantages and disadvantages of the different methods will be discussed. Furthermore, by means of the literature review, the literature where similar OAS approaches have been applied is examined. Relevant literature on the proposed two-step planning approach will be discussed. This question will be answered in Chapter 3

*3. How should the workload optimization, so-called WHAT, model be adapted to generate the realistic scheduling problem instance? How can the highly complex production scheduling problem of AME be decomposed by employing Column Generation, and how should the smart enumeration in the form of branch-and-price be developed?*

Based on the literature review of the previous sub-question, it has become clear what the options are in defining the branch-and-price framework. Answering this sub-question will lead to the definition of the branch-and-price framework for the specific AME production process case. The column generation formulation is examined, the branching method selected, and the enumeration technique determined. After answering this sub-question it is clear how the workload optimization model and branch-and-price framework for the production process of AME are implemented. This question will be answered in Chapter 4

*4. How does the workload optimization model perform in terms of solution quality? How does the resulting branch-and-price framework perform in terms of solution quality?*

In sub-question 4, First, this sub-question is to evaluate the performance of the production target model. Second, to evaluate the performance of the branch-and-price framework. This will be done through a sensitivity analysis. Different instance sizes will be examined to determine the maximum capacity of the model. The model will be tested for different instances to display possible performance differences. The results will be compared to the current scheduling performance to validate the quality. This question will be answered in Chapter 5

# Chapter 2

# Application background: Production system AME

First, the AME production process will be explained in more detail to get a better insight into the production process. Each work center will be defined with the requirements to achieve a realistic representation of the process. Based on the general description of AME's job shop production process and the detailing of the work centers, it becomes clear what kind of production process is involved so that the literature that best fits AME's production process can be identified. Then, in the next section, the relevant literature is discussed.

## 2.1 Production System Description

Each manufacturing company has its own production system descriptions. The purpose of this section is to explain the used production system terminologies to get a better understanding of the system structure.

**Work Centers** AME is an independent developer and manufacturer of high-quality electronic products AME strives for the best solution combining the disciplines of electrical, mechanical, software, and industrial engineering. These disciplines are therefore all realized in-house. AME makes all the parts in-house. The plastic components are produced in-house, the printed circuit boards are soldered and the products are assembled. These different processes are performed in different departments. These departments are described as work centers. Each work center will be explained in more detail in the Section 2.2.

**Resources** Resources can be both people and machines. For each operation type, a set of resources can execute the operation type. Each resource is identified with a unique name. Each machine is committed to one or more operation types the machine can perform. The machine is dedicated to a single work center. The number of hours per day the resource is available is specified for each day. For each operation, a set of identical resources are available that can perform the operation type. It is assumed that these identical resources have equal processing and set-up times per operation. Because these are identical resources, the resources are operational for the same period per day. However, it can happen that an employee is not available all day.

**Products** Each product is related to one work center. These unique products are intermediate products or final products. The products are determined as final when the product is purchased by a customer. Each product has a unique product number (PN). The products consist of several sub-products. It may be the case that several identical sub-products are required for a product. For this purpose, a required quantity of sub-products is specified for the product. These sub-products must be present before the product can be produced in the work center. This is therefore a predecessor priority relationship. It may be that several sub-products are needed to produce a product. However, it is not possible for a product to have more than one successor relationship in the same production order. A predecessor product can be requested from a different work center. To produce a product, a set of sequential operations must be performed. Each product is unique. However, often the same resources are used to produce different products. For example, two different electronic products both contain a printed circuit board. The boards are configured differently because different components are placed. However, both products are processed by the same set of machines, because both products must be marked, components must be placed and both must be soldered. These umbrella processes are revered to as **operation types**. However, the different products are not produced in the same way on the respective machines because the number of components and the type of soldering, for example, are different. These unique processes related to the product are described as **Operations**.

**Orders** AME applies an MTO production strategy, where products are customized to the customer's specifications. As a result, the products of the different customers are unique. Each customer orders a quantity of a specified product number (PN). These products are further defined as a final product because it concerns the final product delivered to a customer. An order consists of a set of jobs that must be performed to produce the order. A job represents the execution of an operation for a batch. Jobs have deterministic processing times. A job can be performed on a set of alternative machines where the production times are the same for the different machines. A job can have precedence relationships, in which there may be one or more predecessors and at most one successor. When production of a job start, the production cannot be interrupted. Preemption is not allowed.

An order is a request for a quantity of products. In order to produce this order, a number of sub-products are needed. An example is given to describe this situation. A customer wants an order of 10 products A. Product A needs two products B to be produced. So this means that the order consists of 10 products A and 20 products B. And the product can be produced by performing a set of operations. A final job represents the group of 10 products A. For this job, the production time is calculated by multiplying the processing time of the operation to be performed by the quantity plus the setup time of the batch. The predecessor job is the production of 20 products B. If a product is produced by performing several operations each operation represents a unique job. A distinction is made between sequential orders and in-tree orders. An in-tree order is an order where at least one of the jobs has multiple predecessor jobs. An example of an in-tree order is presented in Figure 2.1. An order is sequential if all jobs of the order have at most 1 predecessor.

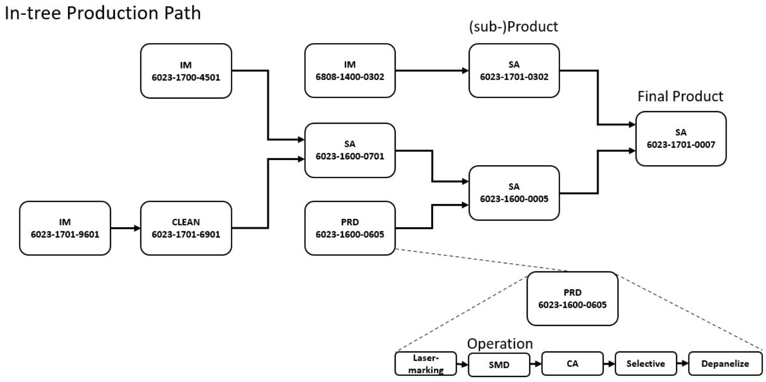



Figure 2.1: In-tree production Path

## 2.2 Work Centers

In this section, the different work centers will be discussed to give indication of the production process and the performed operations.

### 2.2.1 Electronics Manufacturing

In the Electronics manufacturing (PRD) work center, the printed circuit boards are manufactured. The manufacturing of PCBs consists of a set of operations that are manufactured in a fixed order. Depending on the product, certain operations are carried out. The processing time, start-up time, is specified per operation. Each operation can be performed by a set of alternative machines. The processing times are the same for the alternative machines. These alternative machines are assumed to have a fixed and equal operative time per day. However, it can be the case that a machine is not available due to maintenance. This is indicated in the availability per day.

In the Electronics Manufacturing Department, the following operations are carried out:

*Laser marking* - The process starts with a bare PCB. Meaning that no components are mounted on the board. The PCBs arrive in boards where several PCBs are mounted on a board. One or more circuit boards are mounted on each board. The first step is to mark the circuit boards. Each board gets a QR code which is marked with a laser marker. Two alternative identical machines are available to perform this operation type.

*Surface mount devices (SMD)* - The next step in the process is attaching the small components to the PCB. This is done on a surface-mount device (SMD). This is a production line where small components are placed on the PCB by means of an adhesive. The boards are placed in the machine, then glue is applied to the locations where the components are placed. Next, the components are attached to the PCB by means of a machine. The components are delivered in rolls on which several of the same components are placed. These rolls are supplied and placed in the machine by an operator. When all the components have been placed, the boards go through the oven so that the glue can harden. During the process, several quality time controls are carried out. If it turns out that the components have a deviation, it is kept separate at the end of the process. There are three identical SMD production lines where the components can be placed.

In the case that no further components need to be placed, the next operation is the depenalizing operation. In the case that conventional assembly is still to be carried the next operation is conventional assembly. If the product needs in-line programming this operation is done after the SMD operation.

*In-line Programming (IPS)-* if some components placed in the SMD line need programming. The component is programmed in this operation. There is one machine able to perform this operation.

*Conventional Assembly (CA)-* Parts required to be soldered on the bard are placed on the board. The Conventional assembly (CA) operation is a manual operation. This is a preparatory stage where components are placed on the board manually or automatically with a machine. In the next Selective Operation, the components are then soldered to the PCB. This operation can be performed either manually or by machine. In total, there are nine workstations where this operation can be performed.

*Selective -* After the components have been placed on the board in the CA operation, the components can be mounted mechanically by a soldering attachment in the Selective Operation. Two alternative machines can perform this operation. After the Selective operation, a Through Hole Inspection is carried out to validate the qualities of the soldering attachments. The selective machine can solder components on one side of the PCB. If the product has components attached on both sides, the board must go through the CA, Selective, and Through Hole Inspection operation again.

*Through Hole Inspection -* After all the components have been soldered, an inspection is required to check whether the soldering has been done properly. If an error is detected, the component is sent to a repair station. The Through Hole inspection workstation is directly placed after the Selective production line.

*In-Circuit Testing (ICT)-* Some customers require the printed circuit boards to be tested in advance. If the customer requires this, the In-Circuit Testing (ICT) operation must be carried out. This is done after all the components have been placed. This operation can be performed in two alternative machines.

*Kitting -* Components soldered to the PCB may be large, and the solder attachment may not be strong enough to hold the component in place. To strengthen the fixation, an additional kit is applied to hold the component in place. There are two alternative machines where this operation can be performed.

*Depenalize -* Once all the components are in place, the PCBs can be removed from the boards. Two alternative machines can perform this operation type.

*Boxing -* After the PCBs are depenalized, the PCBs can be placed in crates. This is the last operation and the product is finished. The product can either be a final or intermediate product.

Depending on the product, a set of operations have to be executed. This is done in a fixed order. This sequence is shown in the flow chart presented in Figure 2.1.

Table 2.1: Operation Sequences PRD

| **Operation Sequence** | **Occurence** |
|---|---|
| [Laser Marking, SMD, CA, Selective, Depanelize] | 43% |
| [Laser Marking, SMD, Depanelize] | 22% |
| [Laser Marking, SMD, CA, Selective, CA, Selective, Depanelize] | 7% |
| [Laser Marking, SMD, CA, Selective, ICT, Depanelize] | 5% |
| [Laser Marking, SMD, CA, Selective, CA, Selective, Kitting, Depanelize] | 4% |
| [Laser Marking, SMD, CA, Selective, Coating, Depanelize] | 4% |
| Other Routings | 15% |

### 2.2.2 Injection Molding

In the Injection Molding work center, plastic parts are produced via a mold. Plastic pellets are heated to a highly plastic state and forced to flow under high pressure into a mold where it solidifies. AME has five molding machines, which vary in size. The size of products affects the size of the mold. When the size of the product increases, more material is required. Per product, it is specified on which machine the product can be produced. The size of the product influences the required amount of raw material and is therefore also influencing the batch sizes.

The name of the machines represents the number of tons of raw material the machine can work with. In total there are five injection molding machines: 80T, 120T, 120T/2k, 300T, and 500T. The products are categorized into seven groups. Each unique group has a different set of alternative machines that can produce the products. Table 2.2 represents the alternative groups. Each of these groups is handled as a unique operation.

The shape and size of the products affect their manufacturability. Products are categorized into three groups, namely: Automatic production, where the products can be produced 24 hours a day, semi-automatic where the products can be produced 18 hours a day and manual where the products can be produced 16 hours a day.

Table 2.2: Operation Types IM

| **Operation Type** | **Alternative Machines** |
|---|---|
| IM_OP0 | 120T, 120T/2K |
| IM_OP1 | 120T, 120T/2K, 80T |
| IM_OP2 | 120T/2K |
| IM_OP3 | 300T |
| IM_OP4 | 300T, 500T |
| IM_OP5 | 500T |
| IM_OP6 | 120T |
| IM_OP7 | 120T, 80T |
| IM_OP8 | 120T, 120T/2K, 300T |

### 2.2.3 Cable and Wiring

At this work center, the cables and wiring required for the final products are produced. Wires and cables are cut to the required length. This can be done automatically by a machine or manually by an employee. Each job is identified to be either of the automatic or manual production type. The production of a wiring product consists of a single operation that is defined on which machine this product is to be produced. Per operation, the processing time is defined.

### 2.2.4 System Assembly

System Assembly is often the final step of production assembled several components into a product. Making a product involves performing a single System Assembly (SA) operation. This operation is performed by a worker. Each worker assembles the product at a workstation. In total there are 28 workstations where this operation can be performed. The employees work an 8-hour shift. The working day consists of two shifts where the workers take over from each other. So, on average, there are always 28 workers per shift. However, this can vary due to holidays. Employees indicate their availability per day. In the actual planning, the planner takes the skills of the employees into account. However, insufficient data has been recorded on this and it is therefore not considered in the proposed planning model. For now, it is assumed that the employee can perform all operations.

Furthermore, special tools and testing equipment are required to assemble the products. However, the requirements are not yet well documented and therefore cannot be included in the planning. It is assumed that sufficient tools and testing equipment are available. The main factor influencing the schedule for SA is to allocate personnel to production orders.

### 2.2.5 System Assembly in Clean environment

This work center is comparable to the SA work center. However, this process is performed in a sterile environment. This work center is further referred to as CLEAN. The capacity is lower compared to the capacity of SA due to fewer orders of products requested by the customer, requiring to be produced in a sterile environment. Just like in SA, the employees work in two shifts per day. Each shift lasts eight hours. Each shift employs 3 workers.

## 2.3 Current scheduling approach

The generation of the current production planning is conducted in several steps. First, the master planner determines which orders are to be produced. Each work center has an additional planner who will then generate a more detailed planning for the corresponding work center based on the master planning.

There is the senior production planner who makes a rough estimate for the entire planning. This planner considers which orders need to be produced in the near future with a time horizon of approximately two weeks. Based on experience, the senior production planner knows roughly what is needed to produce the various order. For the order, the quantity of products is known and based on the deadline, a rough estimate is made of when production should start. The master planner tries to take into account not to overload certain machines and considers orders one by one.

The constructed master planning is then presented to the various work center planners. If the planners of the work center anticipate that the intended planning cannot be achieved, this is communicated back to the master planner and the planning may be adjusted.

### 2.3.1 Current business practice

AME has kept track of the internal delivery performance of each work center handing over work to other work centers. This is represented in table 2.3. It is notable that PRD performs particularly less in meeting its internal delivery deadline. The overall performance, CLIP performance, indicates the percentage of jobs not making the deadline requested by the customer. AME's production performance is relatively poor: on average only 76% of the orders were delivered to the customer on time [6]. AME strives for a CLIP performance of 85%. However, this performance is often not reached. Therefore, AME hopes to improve the scheduling approach.

| **Work Center** | **On-time delivery** |
|---|---|
| PRD | 71% |
| WIRE | 89% |
| IM | 93% |
| CLEAN | 92% |

Table 2.3: On-time delivery between work centers(2020, Q3)

# Chapter 3

# Related work

First, general job shop planning literature is discussed. The different methods are explained here with the advantages and drawbacks of the different methods. Among other things, the reasons for choosing a branch-and-price framework are explained here. Second, production planning methods are discussed that emerged from the literature review with similar characteristics to the AME production process. Similarities and differences with the current literature and the situation at AME are listed. Third, the column generation decomposition techniques of mathematical optimization are discussed. Finally, the literature of the branch-and-price framework is summarized. This is used as the foundation for the model formulation in the next chapter.

## 3.1 Job-shop Scheduling

The job shop scheduling problem is an optimization technique where jobs are assigned to a machine at a specified time. The most common objective is to minimize the makespan. However, other types of objectives are to minimize tardiness or maximizing profit. The job shop optimization problem is often NP-hard.

Because the job shop problem has considerable potential in business, this is a popular topic in literature. There are several types concerning job shop scheduling. In the case of AME, it is flexible job shop scheduling because each job can be performed by a set of alternative identical machines.

The optimization algorithms for scheduling are mainly divided into exact optimization methods and approximate methods. The exact optimization methods include efficient rule approaches, mathematical programming approaches, and branch definition methods. The approximate methods include constructive methods, artificial intelligence, local search, and meta-heuristic algorithms [7].

An overview of the considered scheduling approaches is represented in Table 3.1

The first mathematical models concerning job shop schedules were initially studied by Johnson in 1954 [8]. However, these methods are not good at solving major problems. A branch-and-bound algorithm is a mathematical optimization technique executing a systematic enumeration of candidate solutions using a state-space search. However, for large instances, the solution area becomes large and therefore takes longer to calculate [9].

The exact optimization methods, such as efficient rule-based approaches, mathematical programming approaches, and branch definition methods, can achieve the exact optimum solution in polynomial time for specified job shop scheduling problems. But only small-scale problems can be solved with the exact optimization methods. As the best exact optimization methods for more complex problems, the branch-and-bound methods can obtain the optimal solution in theory, but it is difficult to have real practical applications because of their complexity [7].

The priority dispatch rule method uses priority rules such as shortest processing time, earliest delivery time, or selection of the same machine. These priority rules are computationally easy to compute. However, with this method, there is no guarantee that the solution is of reasonable quality [10]. With insert algorithms, a pre-selection is made which jobs are scheduled. As a result, the model only takes a small portion of the jobs, making it easier to solve the problem. The resulting schedule of this subset is fixed in the master schedule after which new jobs can be added at the available spots of the schedule. However, there is no guarantee that the generated result will lead to an optimal solution as only a small fraction of the jobs are scheduled each time [11]. The bottleneck procedure takes into account the bottleneck machines. When applying a neural network, a machine learning model is trained to make the planning. The model is trained on train data and then applied to the test data. This requires the arrival distribution of orders. The risk when using a neural network is that the model might overfit the test data [12].

Meta-heuristics are strategies that guide the search process through the search space. The goal is to explore the search space more efficiently. Examples of meta-heuristics are the Tabu search method, genetic algorithms [13] and nature biological based algorithms such as ant colony algorithms [14] and particle swarm algorithms [15]. Meta-heuristics have been applied extensively to solve production scheduling problems because of the promising results compared to standard heuristics. However, designing a good heuristic requires a lot of domain knowledge and is very depending on the type of problem. Furthermore, it is difficult to validate the quality of the solution, as the method is not exact.

Van den Akker applied column generation successfully in job scheduling for common due date, parallel machines, and single machines [16][17]. A disadvantage of column generation is that the solution consists of fractional values. This makes the solution possibly infeasible. Achieving an integer solution is by itself an NP-hard problem.

Branch-and-price is an extension of the branch-and-bound method and column generation, and is better suited for solving large problems. The advantage of branch-and-price compared to meta-heuristics is that the quality of the solution can be validated. With meta-heuristics there is a risk that the model remains stuck at a local optimum. For this reason it was decided to continue with the branch-and-price method.

Table 3.1: Scheduling approaches for job-shop scheduling problems

| Group | Method | Literature |
|---|---|---|
| Efficient rule approach | | [8] |
| Mathematical programming | | [18] |
| Branch-and-bound | | [19],[20], [9] |
| Column Generation | | [21], [17] |
| Branch-and-price | | [22] |
| Constructive methods | Priority dispatch rule | [10] |
| | Insert algorithm | [23] |
| | Botleneck based heuristics | [24], [11] |
| Artificial intelligence | Neural network | [12] |
| Local search | Simulated annealing | [25] |
| Meta-heuristic | Genetic algorithm | [13] |
| | Tabu search | [26] |
| | Ant colony algorithm | [14] |
| | Particle swarm algorithm | [15] |
| | Firefly algorithm | [27] |

## 3.2 Recently conducted studies at AME

The scheduling is based on the two-level scheduling approach. The what-model determines which orders have to be produced, which orders are canceled, and the intermediate deadline per work center. The how-model focuses on how to schedule the projects in order to minimize the deadline violation of orders.

In an earlier study, Firat and Meyere developed a production planning model, the what-model, to pre-determine required deadlines and sub-deadlines in order to minimize the number of tardy orders [5]. The study proposes a Mixed Integer Linear Programming model that converts the customer non-rejected orders into feasible workloads of production units by possibly delaying some of their deadlines. Consequently, the overall tardy jobs will be minimized. However, the company cannot implement this tool immediately because only the deadlines are determined, and no actual schedule is produced. This model will be used as the what-model.

As a follow-up study, Voorn formulated a column generation framework[28] to construct the production schedule which minimizes the number of tardy jobs. This production planning model was intended to use as the how-model in the two-step order acceptance and scheduling approach. However, the results were not as desired and this work is therefore not considered in this study. The pricing problem generates a schedule per work center by performing different heuristics. Per work center, various heuristics were applied, the heuristic with the minimum tardiness is then selected. Heuristics applied are Earliest Due Date (EDD), Shortest Processing Time (SPT), Apparent Tardiness Cost (ATC), Weighted Shortest Processing Time (WSPT), and Minimum Slack (MS).
The master problem selects a schedule per work center. The master problem uses LP relaxation to reduce the complexity the binary values in order to solve the model more efficiently. This leads to fractions in the outcome of the objective solution and is, therefore, an infeasible solution. The column generation model did not result in integer solutions and therefore the results were not meaningful. The best outcome of Voorn's work was the solution to the greedy scheduling model

used to generate basic schedules per work center which were then further explored by means of column generation. However, the result was not satisfactory in terms of solution quality because there are still many orders that did not meet the deadline.
There were two drawbacks to Voorn's implementation. First, a heuristic cannot guarantee that the pricing problem is optimally solved. A second disadvantage is that by solving the planning per work center, the precedence relations still have to be taken into account between the different work centers, because there are precedence relations between jobs of an order which may have to be executed in different work centers.

## 3.3 Order Acceptance Scheduling

Over the past 20 years, the topic of order acceptance has attracted considerable attention from those who study scheduling and those who practice it [29]. Numerous different approaches can be taken to optimize the production scheduling process. The production scheduling model created in this thesis is not a completely new approach, but contains several different caracteristics that have not been applied or combined before in the literature. The purpose of this chapter is to describe different approaches that have been adopted in the literature to solve similar problem, as well as to understand which aspects of the created model have not yet been addressed in previous research.

The production process of AME is a job shop scheduling problem, with the objective of minimizing tardiness of orders. Orders consist of a set of jobs in which there are precedence relations. Jobs have deterministic process times, where the set-up times of jobs must be taken into account. When starting to produce a job, the job cannot be interrupted. Preemption is therefore not allowed. A job can have several predecessor jobs, this creates the in-tree order structure. Furthermore, the model must be able to solve the model in a reasonable computation time. In addition, some listed methods are not capable of solving size problems.
Table 3.2 shows OAS approaches, which were obtained from the literature study, with their corresponding characteristics.

The existing literature does not fully match the characteristics of the AME production process. The papers listed are all theoretical models but have not been applied as case studies. Furthermore, the models in the literature are tested on smaller instance sizes and it is therefore questionable whether the approaches can handle the instance size of the AME case study. To the best of our knowledge, there is no literature available that directly relates to the production situation of AME and for this reason is highly relevant, because it connects to a real-world production process.

Table 3.2: Relevant OAS scheduling Literature

| Year | OutAuthors | Objective | Method | Deterministic | Setup | Preemption | Precedence relations | In-tree handling | Orders | Jobs |
|---|---|---|---|---|---|---|---|---|---|---|
| 2020 | li, Ventura & Bunn [30] | Max revenue | Graph based DP | x | x | x | | | 50 | 50 |
| 2011 | Mesty, Damodaran, & Chen [22] | Max revenue | B&P Graph based | x | x | x | x | | 10 | 100 |
| 2016 | Wang & Ye [31] | Max revenue | MILP | x | x | | | | 50 | 50 |
| 2001 | Hans [32] | Min tardiness | B&P Graph based | x | x | x | x | | 10 | 50 |
| Case study: AME | | Min tardiness | B&P graph based | x | x | | x | x | 40-90 | 170-400 |

## 3.4 Column generation

For a detailed taxonomy of the column generation literature is refer to the work of Desaulniers, Desrosiers and Solomon [21]. In the following, the principle of column generation will be explained. This is based on the work of Desaulniers, Desrosiers and Solomon.

Column generation is an iterative solution procedure for LP models that have a large number of variables. All these variables should not be considered initially, hence column generation starts with some, possibly empty, set of decision variables in the master problem. The new promising columns are generated by solving the pricing problem whose objective is formed with the dual information provided by the master solution. The procedure is terminated when no promising columns can be found any more. This brings the evidence of having optimally solved the master problem.

## 3.5 Branch-and-price

The following is a description of the branch-and-price based on the paper of Barnhart's [1]. Branch-and-price is a branch-and-bound search where a specific lower bounding technique is used, namely Column Generation. The benefit of branch-and-price compared to branch-and-bound is that it is more efficient in solving large problems. Applying a standard branch-and-bound procedure to the master problem over the existing columns is unlikely to find an optimal, or even good, or even feasible solution to the original problem [33].

The ILP master model is relaxed to be solved optimally in polynomial time by employing the Column Generation. However, the integrality property remains to ensure, since the optimal solutions of LP models are in general fractional. Then branch-and-price comes into games as a smart enumeration method to explore the search space.

Branch-and-price makes use of the property that most of the columns are equal to zero, and therefore do not contribute to the optimal solution. Therefore, it is not required to add all columns to calculate the optimal solution. By not including all columns at once, the computational power and memory requirement are reduced.

At the start of the calculation, only a few columns are added, this is called the Restricted Master Problem. Next, the subproblem (pricing problem) is solved to identify columns that could be included to improve the solution. The pricing problem is a separate problem from the dual LP.

If there are columns that have a negative reduced cost for a minimization problem, it means that adding this column has high potential for decreasing the objective value. The newly found columns are added and the restricted master problem is resolved. As long as new promising columns are found, this iterative process continues. There are several approaches to find the columns to be added. Heuristics and local search methods can be applied depending on the problem. Further research is required to determine the best suitable type of column generation for this specific problem.

If there are no columns found to be added to the Restricted Master problem. The solution is optimal. However, due to the LP relaxation, the solution could consist of fractional values instead of an integer value. Therefore, the solution is not in the desired form. Branching needs to be applied in order to get rid of the fractional values in the solution. This branching consists of two rules, namely the node selection rule and branching rule. The node selection rule determines in which order the nodes are treated. Some standard rules are [34]:

- *Depth-first search* in which one first elaborates on a single branch. This method leads faster to a result compared to breadth-first search. But it has an increased risk of not reaching a sufficient quality if an incorrect branching method is applied.
- *Breadth-first search* in which one selects nodes that are closest to the top of the tree. Breadth-first search has a smaller risk of not getting a sufficient quality because the branch is worked out gradually between all branches. However, this method is rather slow because it is likely to take longer to find an integer solution.
- *Best-bound search* in which a node is selected based on the most promising lower bound.
- *Combination* a combination of breadth-first search and depth-first search. This search attempts to get a compromise of both approaches. Meaning the computation speed of depth-first search and the robustness of breadth-first search is combined. However, the implementation is highly dependent on the problem.

The branch rule determines which fractional variable to select for branching. A common approach is selecting the most fractional number, so closest to 0.5, because this variable is the furthest from an integer solution. Another approach is determining a branching priority because some variables are, for example, more likely to have a major influence on getting an integer solution. However, this is case-dependent.

The flow chart of the branch-and-price algorithm is shown in Figure 3.1.

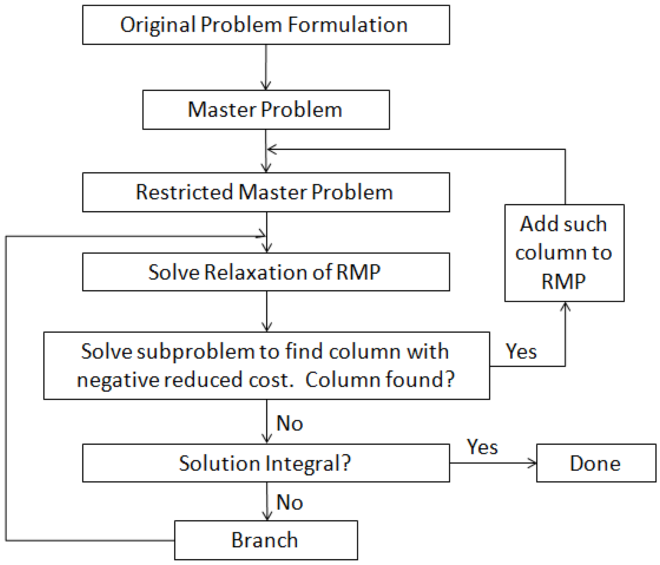



Figure 3.1: Branch-and-Price algorithm [1]

# Chapter 4

# Methodology

The previous section explained the components of the branch-and-price framework. Based on the production characteristics mentioned in Chapter 2, the mathematical formulation of the branch-and-price framework can now be further explained. First, the production target model will be explained and then the planning model will be discussed.

This study is an extension of the research by Firat and Meyere [5]. In this study, a model was developed for the what model. This will also be used in this study to determine the production target. However, some adjustments have been made to this model because the model developed in the earlier study did not take into account resource availability. The revised model is further explained in Section 4.1. Section 4.2 then elaborates on the how model.

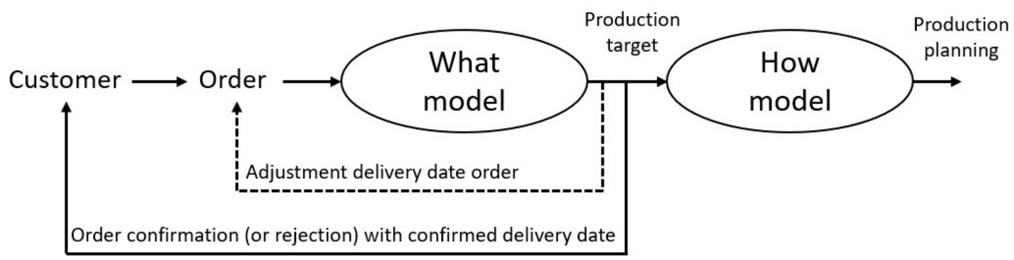



Figure 4.1: Overview two-level production planning

## 4.1 Step one: Order acceptance

The workload optimization model is an extension of the work of Firat and Meyere [5]. This model unifies the order-selection decision problem along with the aggregate production-planning problem through setting system-aware production targets and controlling the workload that enters the system [5]. The model determines which orders to accept and the required deadline. If deadlines determined by the customer are infeasible, the deadline can be adjusted by the model. In the case of two many orders resulting in not enough machine capacity, the orders will be rejected.

First of all, a planning horizon is defined which determines which orders the model will take into account. When the deadline falls in the planning horizon the entire order is included in the model. Furthermore, for each order, the latest possible start time is determined in which the deadline can still be met. If the deadline of an order falls outside the time horizon, but the latest start time of the order falls in the planning horizon the order is partially included. This approach for defining the considered orders is represented in Figure 4.2. The deadline is adjusted to the planning horizon and the corresponding order quantity is adjusted. The adjusted quantity is calculated by taking the fraction of the order that can still be produced before reaching the time horizon.

### Problem description

This section formulates the basic notation used for the workload optimization model. The model depends on three different objects, namely: Production units, product types, and customer orders. These objects are explained in Section 2.1. In the follows the mathematical notation is defined.

The time horizon is defined as $\tau$ (Figure 4.2), in days. The production units are a set of resources $m \in R$. Each resource has a production capacity, in minutes, $H_m$ specified for each resource $m$. Each resource has a maximum target utilization $U_m$. The set of products $i \in P$ are the intermediate or final products. For each product, there is a processing time $p_i$ defined, a setup time $stp_i$ for a batch, and an estimated batch size $B_i$.

For the set of Orders $o \in O$, a deadline $d_o$ is defined for each order with the customer's desired deadline. The latest start time $s_o$ indicates the latest possible start time for an order to fin-

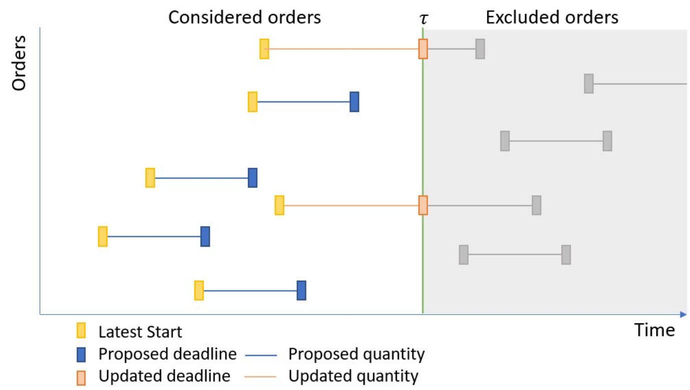



Figure 4.2: Approach for defining considered orders in Workload Optimization model

ish production just before the desired deadline. The maximum customer tolerance $C$ presents the delay which is tolerated by the customer. The deadline of order $o$ can be delayed between $[d_o, d_o + C]$ that is indicated by one of the $\theta$ variables. If the order needs to be delayed more than $C$ then the order must be rejected ($r_o = 1$).

The order quantity $q_o$ presents the number of final products of order $o$. If the deadline of the order is after the time horizon $\tau$ the quantity is updated. The updated quantity is calculated as:

$$q_o^* = \begin{cases} q_o, & \text{if } d_0 \leq \tau \\ \frac{d_o - s_o}{\tau - s_o} \cdot q_o, & \text{if } d_o > \tau \end{cases}$$

$\alpha_{i,o}$ indicates the number of sub-products required to produce one final product of the order. The necessary flow time in the system from completion of component $i$ till completion of one final product required by order $o$ is denoted by $F_{o,i}$. The desired production quantity for a product is defined as $\kappa_{i,t}$ for product $i$ at day $t$.

In order to capture the capacity demand of products that can be produced in alternative machines, the definition of an individual machine is generalized to a subset of machines. This way the overlap in alternative machines of products can be better handled in capacity constraints.

**Machine subsets as resources** The set of all subsets of machines, $\mathcal{R}$, covers all possible unions of job alternatives machines. The idea of using these subsets is to capture the capacity use of machines in a work center for the jobs that have alternative machines. $\mathcal{R}$ has the following property

$$R_j \in \mathcal{R} \text{ and } R_j \cup R_{j'} \in \mathcal{R}, \quad j, j' \in \mathcal{J} : R_j \cap R_{j'} \neq \emptyset \tag{4.1}$$

An example is given to clarify the machine subsets. Suppose there are 3 different products. Product 1 which can be performed on machine 1 and machine 2, $R_1 = \{M1, M2\}$, product 2 can be performed on machine 2 and machine 3, $R_2 = \{M2, M3\}$, and product 3 can be performed on machine 3 and machine 4, $R_3 = \{M3, M4\}$. Based on the 3 products, the following subsets are created:

$\mathcal{R} = \{\{M1, M2\}, \{M2, M3\}, \{M3, M4\}, \{M1, M2, M3\}, \{M2, M3, M4\}, \{M1, M2, M3, M4\}\}$.

**Latest start times of orders** The original model, created by Meyere [6], made some unrealistic assumptions regarding defining the latest start time of orders. As a result, the model made inaccurate estimates of the feasibility of orders. The original model used the assumption that the latest start was always two weeks before the order deadline. However, order sizes vary and therefore this is an unrealistic assumption. To improve the accuracy, a recursive back-propagation through the production path is used. Based on the desired quantity and the required processing time per product, the actual latest start can be calculated accurately.

In order to generate a realistic production target, it is imperative that the model includes resource availability in determining the production target. This was not yet incorporated in the original model of Firat and Meyere and has therefore been revised. Furthermore, the model has been further optimized, a number of constraints have been rewritten for better computational efficiency.

Table 4.1: Sets, Parameters, Decision Variables for workload optimization model

| | |
|---|---|
| ***Sets*** | |
| $O$ | set of all customer orders, indexed by $o$. |
| $P$ | set of all components and product types, indexed by $i$. |
| $R$ | set of all resources (machines), indexed by $m$. |
| $R_i$ | set of resources $m$, on which product $i$ can be performed. |
| $\mathcal{R}$ | all sets of machines that can produce common components. |
| $T$ | set of all days, indexed by $T \in \{0, 1, ..., \|T\|\}$, |
| $T_o$ | set of all considered days for target completion, indexed by $T \in \{d_o, ..., d_o + C\}$, |
| $P_o$ | set of components required for final product of order $o$, $P_o \subset P$ |
| $P_m$ | set of components produced in production resource $m$, $P_m \subset P$ |
| ***Parameters*** | |
| $\mathbf{M}$ | big-M parameter. |
| $q_o^*$ | Updated order quantity of the final products of order $o$ |
| $s_o$ | latest production start of order $o$. |
| $d_o$ | deadline of order $o$. |
| $p_i$ | processing time of item $i$. |
| $\alpha_{i,o}$ | number of sub-products $i$ required to produce one final product for order $o$. |
| $U_m$ | utilization ratio of the resource $m$ . |
| $A_{m,t}$ | Availability of machine $m$ at day $t$. |
| $stp_i$ | setup time required for a batch size of component $i$. |
| $C$ | customer order delay tolerance. |
| $F_{i,o}$ | flow time of item $i$ to become a final product of order $o$. |
| $B_i$ | estimated batch size of item $i$. |
| $H_m$ | length of one workday for resource $m$. |
| $l_i$ | Initial buffer level of component $i \in P$ |
| $\tau$ | time horizon |
| ***Decision variables*** | |
| $\theta_{o,t}$ | binary indicating that target completion data is $t$, possibly by delaying the deadline $d_o$. |
| $I_{i,t}$ | required number of item $i$ at time unit $t$. |
| $r_o$ | binary indicating order $o$ is rejected. |
| $\phi_{i,t}$ | the number of setups necessary for producing component $i$ by time $t$. |

**Workload optimization model formulation** This section defines the formulation of the workload optimization model. Table 4.1 introduces the sets, parameters, and decision variables used in the workload optimization model.

$$\text{Minimize} \sum_{o \in O} \Big( M r_o + \sum_t (t - d_o)^+ \theta_{o,t} \Big) \tag{4.2}$$

subject to

The objective in (4.2) is primarily minimizing the number of rejected orders and secondarily the delay between original deadlines and target delivery dates.

$$\sum_{t \in T_o} \theta_{o,t} + r_o = 1, \quad o \in O \tag{4.3}$$

The constraints (4.3) ensure that an order is either rejected ($r_o$) or accepted ($\theta_{o,t}$) when the time horizon ($\tau$) passes consideration time of order $o$.

$$I_{i,t} = I_{i,t-1} + \sum_{o \in O : t = t' - F_{i,o}} \alpha_{i,o} q_o \theta_{o,t'}, \quad t \in T, i \in P \tag{4.4}$$

$$I_{i,t} \leq l_i + \kappa_{i,t}, \quad t \in T, i \in P \tag{4.5}$$

Constraints (4.4) determines the number of products $i$, required at day $t$. The constraint takes into account the total required products and the quantity produced in the past and the required quantity to be produced at day $t$. In the case of stock of products, this is taken into account in constraint (4.5). This constraint updates the required products to be produced based on the initial stock of the component.

$$\phi_{i,t}B_i \geq \kappa_{i,t}, \quad t \in T, i \in P \tag{4.6}$$

$$\sum_{i \in P_r}(stp_i\phi_{i,t} + p_i\kappa_{i,t}) \leq \sum_{m \in r}\sum_{t'=1}^{t} U_m H_m A_{m,t'}, \quad t \in T, r \in \mathcal{R}, \tag{4.7}$$

$$\theta_{o,t}, r_o \in \{0, 1\} \quad t \in T_o, o \in O \tag{4.8}$$

Some production orders are too large to produce in a single batch. As a result, multiple set-ups are required to produce the order. Constraint (4.6) defines the required number of batches to produce the desired quantity of product $i$ per day. Constraint (4.7) ensures that the machines do not produce longer than the available time per day. The sum of the product production times together with the required setup times of all the products cannot be more than the available processing time over the alternative machines in the machine subset until day $t$.

The model respects the capacity of machines and determines target deliveries of orders, possibly by delaying the original deadlines within customer tolerance. If the production of the order is still unfeasible, the model can choose to reject the order. The result of the model is a list of accepted orders. It is possible that the quantity has been adjusted because the original deadline falls outside the time horizon or the deadline has been delayed in the range of the original deadline and the original deadline plus the customer's maximum tolerance for which the order can be delayed.

The results of the production target model are presented in Section 5.3.

## 4.2 Second step: Production Scheduling

The second step of the approach is scheduling the jobs that are determined by the workload optimization model. This scheduling problem is a flexible job shop scheduling problem with precedence relations of parallel jobs of final products. This section formally defines the scheduling problem and presents the master and pricing problem formulations respectively.

The production scheduling problem is solved through a branch-and-price framework. The mathematical formulation is described in this section. First, the problem definition will be given. Secondly, the column generation formulation is explained. Finally, the further formulation of the branch-and-price framework is explained with the node selection method, branching method, and method to generate the initial feasible schedule for the root node.

### 4.2.1 Problem definition

The proposed production scheduling approach is a flexible job shop scheduling problem with multiple alternative machines. Let $o \in O$ be the set of all orders with deadline $d_o$. Each order

$o \in O$ requires a set of jobs $J_o$ in certain precedence relations to be scheduled. Given a job $j \in J_o$ of order $o$, the processing time $p_j$ is assumed to be deterministic and it also includes the setup time. A job $j \in J_o$ can be performed on a set of alternative machines $m \in M_j$.
It is assumed that the processing time of a job is the same on all alternative machines of that job. The predecessors and successors of job $j$ are denoted by $Pred_j$ and $Succ_j$ respectively. The processing routes of final products have an in-tree structure. This means that $|Pred_j|$ may be greater than 1 and $|Succ_j| = 1$ for every job $j$, except the final products.
The time horizon is indicated by $T$, in days. $T$ is big enough to allow feasible schedules, hence the optimal one. A time granularity $g \in G$, in minutes, is introduced as a time interval between two points in time when a job could start. The process times are rounded up to a multiple of the time granularity. The set of time points when a job can start is defined as $\{0, g, 2g, 3g, ..., T \times 1440\}$, where 1440 is equal to a multiple of $g$. The objective of the scheduling problem is to minimize the tardiness of orders. The tardiness of an order is defined based on the completion time of the final job $j^*$ and the order deadline $\tau_o = (c_{j^*} - d_o)^+$.

### 4.2.2 Order based formulation of the scheduling problem

This section explains the reformulation of the scheduling problem using order schedules as decision variables. The model considers the feasibility of machine use and selects one schedule for every order.

The advantage of the reformulation based on schedules of orders is that this way of formulating the problem eliminates the handling of precedence relations by the master problem since there are only precedence relations in orders. As a result, the master problem only has to take into account the machine occupancy, since the machines cannot process two different orders simultaneously. The notation is introduced in Table 4.5 and the formulation is given in expressions (4.9)-(4.12).

Table 4.2: Sets, parameters, and decision variables for the order schedule reformulation

| | |
|---|---|
| ***Sets*** | |
| $O$ | set of orders $o \in O$. |
| $\Gamma_o$ | set of schedules of order $o \in O$. |
| $M$ | set of machines $m \in M$. |
| $T$ | set of time units $t \in T$. |
| ***Parameters*** | |
| $y_{o,s,m,t}$ | binary parameter indicating that schedule $s$ uses machine $m$ at time $t$ for processing a job of order $o$. |
| $\tau_{o,s}$ | tardiness of order $o$ in schedule $s$. |
| ***Decision Variables*** | |
| $\lambda_{o,s}$ | binary variable indicating that schedule $s$ is selected for order $o$ |

$$\text{Minimize} \sum_{o \in O} \sum_{s \in \Gamma_o} \tau_{o,s} \lambda_{o,s} \tag{4.9}$$

The objective of the master problem (4.9) is to minimize the sum of tardiness over all orders. For each order, there is a set of alternative schedules $\lambda_{o,s}$. The tardiness of the schedule is denoted by $\tau_{o,s}$.

subject to

$$\sum_{s \in \Gamma_o} \lambda_{o,s} = 1, \qquad \forall o \in O \tag{4.10}$$

$$\sum_{o \in O} \sum_{s \in \Gamma_o} y_{o,s,m,t} \lambda_{o,s} \leq 1, \quad \forall (m,t) \in M \times T \tag{4.11}$$

$$\lambda_{o,s} \in \{0,1\} \quad \forall s \in \Gamma_o,\ o \in O \tag{4.12}$$

Constraint (4.10) ensures a schedule is selected for each order. When the master problem is solved the integer restrictions of the variables are relaxed. As a result, the constraint ensures that the sum of the selected schedules is equal to one. However here the avoidance of double occupancy of the machines must be taken into account. For each alternative schedule of an order is defined when the order is occupying a machine. This is indicated with the parameter $y_{o,s,m,t}$. Constraint (4.11) ensures the utilization is not exceeded for each machine at each time point.

### 4.2.3 Pricing problem

The relaxation of the master model can be solved using column generation by adding promising order schedules to schedule sets of orders $\Gamma_o$ for all $o \in O$. In order to obtain the pricing problem, the reduced cost of $\tau_{o,s}$ is defined as

$$\overline{\tau}_{o,s} = \tau_{o,s} - \pi_o + \sum_{m,t} \pi_{m,t} y_{o,s,m,t} \tag{4.13}$$

where the dual variables $\pi_o$ and $\pi_{m,t}$ are associated with constraints (4.10) and (4.11).
The pricing problem can be defined as finding $z^*_{Pr} = \min\limits_{s \in \bigcup_{o \in O} \Gamma_o} \{\overline{\tau}_{o,s}\}$. It can be seen that the problem of finding $z^*_{Pr}$ can be decomposed into solving $|O|$ optimization sub-problems for each of the orders. More precisely,

$$z^*_{Pr} = \min_{o \in O} \{ \min_{s \in \Gamma_o} \{\overline{\tau}_{o,s}\}\},$$

For each order, the pricing problem is to construct a schedule that minimizes the corresponding objective value. In order to make a systematic search for pricing problem (of order $o$). The binary variable $y_{o,s,m,t}$ indicates that a job of order $o$ is being processed on machine $m$ at time $t$ in schedule $s$, and $c_{o,s}$ as the latest job completion time of schedule $s$. Then the pricing objective becomes

$$z_{Pr}(o) = \min \left\{ (c_{o,s} - d_o)^+ + \sum_{s \in \Gamma_o} \sum_{m,t} \pi_{m,t} y_{o,s,m,t} \right\} - \pi_o,$$

Hence its completion time directly determines the tardiness of the order.

**Formulating the pricing as MILP**

In this section, the formal definition of the pricing problem is given in the form of a MILP model. The Pricing problem is finding an order schedule that is promising for the master problem with the available order schedules in it.

In the pricing problem, jobs of a given order are considered for scheduling to obtain an order schedule. Hence, the notation pricing search includes jobs and related decision variables. So we have $c_{o,s} = \max_{j \in J_o}\{c_j\}$ and $y_{o,s,m,t} = \sum_{j \in J_o} x_{j,m,t}$.

Later the pricing problem is transformed to a graph notation. A distinction is made between sequential and in-tree orders. For the in-tree orders, the pricing problem can be solved using MILP model. In the case of sequential orders, the pricing problem can be solved both with MILP model and a Single Source Shortest Path Problem on a specific graph. The standard notation is introduced in Table 4.3 and the formulation is given in expressions (4.14)-(4.22). The arc-based notation is introduced in Table 4.4 and the formulation is given in expressions (4.23)-(4.27).

Table 4.3: Sets, parameters, and decision variables of pricing problem

| | |
|---|---|
| ***Sets*** | |
| $O$ | set of orders $o \in O$. |
| $J_o$ | set of jobs of order $o$ |
| $M$ | set of machines $m \in M$. |
| $T$ | set of time units $t \in T$. |
| ***Parameters*** | |
| $\pi_{m,t}$ | cost of using machine $m$ at time $t$ for processing a job. |
| $A_{m,t}$ | availability of machine $m$ at time $t$. |
| $p_j$ | processing time of job $j$. |
| $d_o$ | deadline of order $o$. |
| $Pred_j$ | set of predecessor jobs of job $j$ |
| ***Decision Variables*** | |
| $x_{j,m,t}$ | binary variable indicating job $j$ is processed in machine $m$ at time unit $t$ |
| $st_{j,m,t}$ | binary variable indicating processing of job $j$ in machine $m$ started at time unit $t$ |
| $c_j$ | completion time of job $j$ |
| $c_{j^*,o}$ | completion time of final job of the order |
| $\tau_o$ | tardiness of order $o$ |

$$\text{Minimize } (\tau_o + \sum_{j \in J_o} \sum_{m} \sum_{t} \pi_{m,t} x_{m,j,t}) \tag{4.14}$$

The objective 4.14 of the pricing problem is to minimize tardiness and the sum of the used dual weights over all machines, jobs and time points. $x_{m,j,t}$ is a binary decision variable indicating whether the job $j$ is being produced on machine $m$ at time point $t$. The variable $\pi_{m,t}$ is the dual weight of constraint 4.11. This dual how convenient to execute a job on the machine at time point $t$ regarding the selected order schedules in the solution.

subject to

$$\sum_{t} \sum_{m} st_{m,j,t} = 1, \quad \forall j \in J_o \tag{4.15}$$

$$\sum_{m} \sum_{t} x_{m,j,t} = p_j, \quad \forall j \in J_o \tag{4.16}$$

$$x_{m,j,t} \leq st_{m,j,t} + x_{m,j,t_t^o}, \quad \forall m \in M, j \in J_o, t \in T \tag{4.17}$$

$$t(x_{m,j,t-1} - x_{m,j,t} - (1 - A_{m,t})) \leq c_j, \quad \forall m \in M, j \in J_o, t \in T \tag{4.18}$$

$$\sum_t x_{m,j,t} \leq p_j \sum_t st_{m,j,t}, \quad \forall m \in M, j \in J_o \tag{4.19}$$

$$c_{j'} \leq \sum_m \sum_t t \cdot st_{m,j,t}, \quad \forall j' \in Pred_j, j \in J_o \tag{4.20}$$

$$\sum_j x_{m,j,t} \leq A_{m,t}, \quad \forall m \in M, t \in T \tag{4.21}$$

$$c_{j^*,o} - d_o \leq \tau_o \tag{4.22}$$

The first constraint (4.15) ensures a job can only start at the specified time point $t$ on machine $m$. With constraint (4.16) the completion time of a job is set to be equal to the number of binary variables indicating the job is processed. The third constraint (4.17) ensures a binary variable indicating the job is processed can only be equal to one if the previous time point was dedicated to the same job, $x_{m,j,t_t^o}$ represents the off-time. This will ensure that the job cannot be interrupted. Constraint (4.19) ensures the job can only be executed on the machine on which the job started. Constraint (4.20) ensures that precedence relations between jobs are not violated. Here $j'$ denotes the successor of job $j$. A machine can only be used if it is available at time $t$. This is handled by constraint (4.21). Finally there is constraint (4.22), which defines the tardiness based on the completion time of the last job and the deadline of the order.

The pricing problem is calculated each iteration of the column generation and has therefore a large impact on the total computation time. It is important to have a relatively short computation time, and be able to solve the model optimally. For this reason, the model is simplified.

The disadvantage of this method is there are many decision variables that the model has to consider. Namely, there is the start time and all the time points which indicate whether the job will be produced at time point $t$. The decision variable $x$ is flexible in the model while the completion time can already be defined in advance based on the start time since the availability is fixed from the start.

### Arc based presentation of the pricing problem

Instead of fixing the start time and occupation times of a job, another option is to define a set of possible start times for each job, each of the elements of the set is called an arc. An arc represents a unique start time $t$ on machine $m$, where the tail represents the start time of job $j$ on machine $m$, and the head of the arc represents the completion time of the job. To reduce the number of arcs, a time granularity is introduced. This indicates the interval between different time points. Instead of a job being able to start each minute, the job can only start at a multiple of the time granularity. As a result, this reduces the number of arcs by a factor of the time granularity.

$$\text{Minimize } (\tau_o + \sum_{j \in J_o} \sum_{a \in A_j} \pi_a x_{a,j}) \tag{4.23}$$

subject to

Table 4.4: Sets, parameters, and decision variables of arc based pricing problem

| | |
|---|---|
| ***Sets*** | |
| $J_o$ | set of jobs of order $o$ in precedence relations. |
| $A_j$ | set of arcs representing feasible execution of job $j$. |
| $\mathcal{C}$ | set of pairs of arcs which have conflict in machine use. |
| $Pred_j$ | set of predecessor jobs of job $j$ |
| ***Parameters*** | |
| $st_a$ | start time of arc $a \in A_j$, $j \in Jo$ |
| $c_a$ | completion time of arc $a \in A_j$, $j \in Jo$ |
| $\pi_a$ | dual objective coefficient of arc $a \in A_j$, $j \in Jo$ |
| ***Decision Variables*** | |
| $x_{a,j}$ | binary variable indicating that arc $a \in A_j$ is selected. |
| $\tau$ | tardiness of order $o$. |

$$\sum_{a \in A_j} x_{a,j} = 1, \quad \forall j \in J_o \tag{4.24}$$

$$\sum_{a \in A_{j'}} c_a x_{a,j'} \leq \sum_{a \in A_j} st_a x_{a,j}, \quad \forall j \in J_o, j' \in Pred_j \tag{4.25}$$

$$x_{a,j} + x_{a',j'} \leq 1, \quad \forall (a, a') \in \mathcal{C}, j, j' \in J_o \tag{4.26}$$

$$\sum_{a \in A_{j^*}} c_a x_{a,j^*} - d_o \leq \tau_o, \tag{4.27}$$

The first constraint (4.24) ensures that an arc is selected for each job since a job can only be performed once. The second constraint (4.25) ensures the precedence relations are not violated. The third constraint (4.26) ensures the machines cannot be assigned to different jobs at the same time. Here, for each arc $x_{a,j}$, a set of conflicting arcs $x_{a',j'}$ is defined. In the case of a sequential order, the precedence rule ensures that different jobs can never be produced at the same time on the same machine. For jobs of a sequential order, the set of conflicting arcs will always consist of an empty set. However, in the case of an in-tree order, it may be the case that a job has multiple predecessors. If these predecessors can be produced on the same machine, it may be the case that these predecessors are assigned on the same machine at the same time. This double occupancy is prevented by the constrain. Finally, constraint (4.27) ensures that the tardiness of the schedule is adjusted based on the deadline of the order and the completion time of the final job $j^*$.

### Transforming arc based pricing problem to shortest path problem

The pricing problem is transformed into an order execution to facilitate the possibility of solving the problem with a shortest path algorithm. By formulating the problem as a shortest path problem the pricing problem of sequential orders can be computed more efficiently.

The processing of an order is modeled as a directed graph, an example of a directed graph for a sequential order with three jobs is presented in Figure 4.3. The arcs represent either execution of a job (diagonal arcs), waiting idle arcs between executions of two successive jobs (vertical arcs), or the tardiness penalty in the case of the arcs reaching the sink node. Execution arcs of jobs are topologically ordered in the same way as the precedence relations between jobs of an order. There are two distinguished nodes. The node $N_{0,0}$ with only outgoing arcs, the source node (Source in Figure 4.3), and the one with only incoming arcs, the sink node (Sink in Figure 4.3). A (directed) path from Source to Sink is a feasible processing of an order respecting the precedence relations by its structure.

Let $G = (N, A)$ denote the order execution graph, where $N$ is the set of the nodes and $A$ is the set of the arcs. The node $n_{t,j} \in N$ represents the state where job $j$ is executable, at time point $t$. The current time of the time point can be obtained by multiplying the time point by the time granularity $g$. The source node is identified as $n_{0,0}$, and sink node as $n_{sink}$. The sink node has only incoming arcs from nodes $n_{t,|J_o|}$ for all feasible $t$.

Diagonal arc $a^m_{[n_{t,j},n_{t,j+1}]}$ represents execution of job $j$. Vertical arc $a_{[n_{t,j},n_{t+1,j}]}$ represents one time granularity of waiting time before starting processing of jobs $j$. The parameter $p_{t,j}$ represents the execution time for job $j$ including off-times, when the job starts at time point $t$. Here the off-time must be taken into account when the job starts at the end of the day and thus is not yet finished at the end of the day. This off-time is added to the processing time to define the execution time.
The set of time points is presented in as

$$P = \{0, 1, 2, ..., \frac{T \cdot 1440}{g}\} \tag{4.28}$$

and $g$ is always a multiple of 1440. The set of jobs is defined as

$$J = \{0, 1, 2, ..., |J_o|\} : j \to j + 1 \tag{4.29}$$

The weight of the diagonal arcs are defined by the machine-time dual variables as

$$w^m_{(n_{(t,j)},n_{(t+\frac{p_{t,j}}{g},j+1)})} = \sum_{t=t\times g}^{t\times g+p_{t,j}} \pi_{m,t} \quad \forall t \in P, j \in J_o, m \in m_j \tag{4.30}$$

The weight of the vertical arcs are defined as

$$w_{(n_{(t,j)},n_{(t+1,j)})} = 0, \quad \forall t \in P, j \in J \tag{4.31}$$

In the case of arcs from the last executable job to the sink node, the weight of the are is the tardiness of the order. The weight of the sink arcs are defined as

$$w_{(n_{(t,|J_o|)},n_{sink})} = (t \cdot g - d_o)^+ \quad \forall t \in P \tag{4.32}$$

Figure 4.3 shows an example for a sequential order containing three jobs. In the example, job 1 can be executed by machine 1 and machine 4, job 2 can be executed by machine 2, and job 3 can be executed by machine 1 and machine 4. Depending on the execution time of the job, the arrow goes up a number of nodes, since for job 1 the processing time is 60 and the time granularity is 30 minutes the arrow goes 60/30 nodes up and one to the right. Because of the sequential relationship, the structure prevents job 1 and job 3 from being executed at the same time on machine 4. The deadline of the order is after 90 minutes. The sink arcs indicate the tardiness of the order. In the example, the time horizon is 180 minutes. In the implementation, the time horizon is in days, and therefore multiplied by 1440 (Expression 4.28).

**Theorem 1.** *The pricing problem for a given order with sequential job precedence relations is equivalent to solve the Shortest Path problem on the order processing graph in Figure 4.3.*

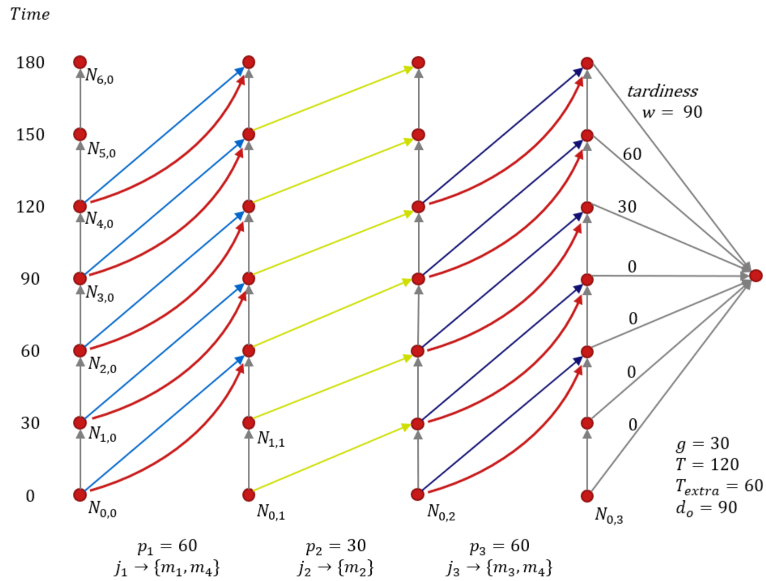



Figure 4.3: Order Graph example

*Proof.* The proof is constructive by showing the coincidence of feasible solutions to both problems. The shortest path from Source node to Sink node an order processing graph in Figure 4.3 gives the optimal pricing solution for order $o$. To show this, a path from the source node to the sink node is a feasible production of an order with exactly the same pricing objective value compared to the MILP model. This is indeed true due to diagonal arc weights. A job execution has the cost of machine use for the time units used. Lastly, the order processing graph contains all possible order processing options by allowing waiting time between job executions. To sum up, finding the shortest path on the order processing graph is equivalent to solving the pricing problem to optimality. □

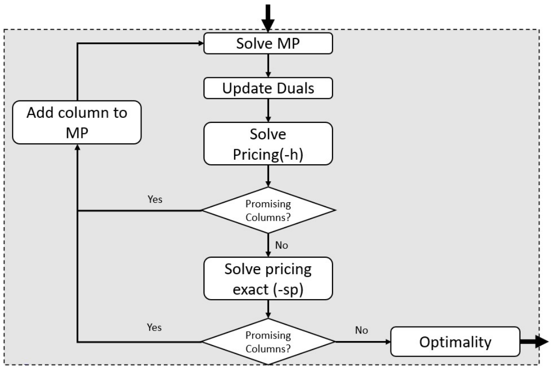



Figure 4.4: Overview Column generation

### 4.2.4 Branch-and-price framework

The branch-and-price framework is based on the proposed format of Barnhart [1]. An overview of this framework is presented in Figure 4.5.

The branch-and-price framework is initialised using the schedules that are constructed by the greedy scheduling method. The generated order schedules from this model are used as an input for the root node of the column generation.

The node can then be explored by applying column generation. The working principle of column generation is shown in Figure 4.4. The model for define the initial solution is explained in more detail in Section 4.2.4. The implementation of the column generation is discussed in section 4.2.3.

After the node is explored, the solution is validated whether it is integer. If not, branch-and-bound optimization is applied to explore the search space of the root node through a systematic enumeration search. The space is recursively split into smaller spaces. The splitting is called branching. In the following of this subsection the branching method is explained (Section 4.2.4).

This process keeps repeating until a node is found with an integer solution within the optimality gap range or when all the nodes have been explored.

This chapter presents how the branch-and-price framework is formulated. Section 4.2.4 further discusses the results of the framework.

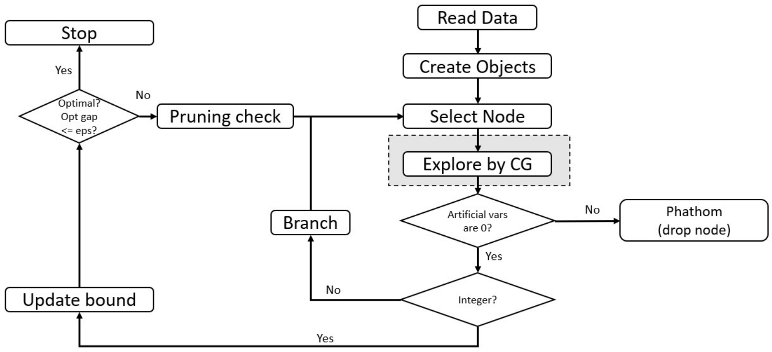



Figure 4.5: Overview branch-and-price structure

#### Initial Solution: Greedy scheduling method for initial schedules

Column generation is well suited to improve a solution. However, it is not well able to generate a root solution. Providing some initial columns that belong to a feasible solution of the master problem is a good hand for the sake of a warm start for Column Generation. Therefore it is useful to generate a basic solution first, which the column generation can further explore. For this reason, the current approach was selected to generate a basic feasible solution.

A greedy model is selected to generate the first feasible solution. The advantage of a greedy model is that it can quickly generate an initial feasible solution without any quality guarantee. Because the job shop problem is too large to be solved directly with an ILP, a pre-selection is made among the jobs to be scheduled. The algorithm starts with empty machine schedules and extends the current (partial) solution by assigning the jobs into machines schedules that are determined by solving an ILP model. In order to solve the ILP model efficiently, a subset of the schedulable jobs is included. This smaller group of selected jobs can then be solved optimally with an ILP model. Each iteration, new jobs are added to the model and additional jobs are scheduled. This process repeats iteratively until all jobs are scheduled. One drawback of this method is that because each time the model solves a small problem, the model does not consider the entire problem. Therefore, this model can not guarantee to generate an optimal solution.

In order to reach a reasonable solution, several criteria are defined based on which the ILP can generate a solution. A grit search is conducted to explore the best parameter set. Firat, Bilge & Albey [35] research defined three main parameters influencing the planning that affect the quality of the solution. These are the earliest possible finish times when the jobs must be completed in order to meet the order deadline, defined as $\alpha$, the number of jobs to be scheduled before the order is completed, defined as $\beta$, and the remaining process times required to complete the order, defined as $\gamma$. Based on these three parameters a normalized weight $w_j$ is defined for each job.

The objective of the ILP model is to maximize the weight $w_j$ (4.33). Each job has a normalized weight $w_j$. Whether a job is assigned to a particular machine is indicated by the decision variable $x_{j,m}$.

$$\text{Maximize} \sum_{j \in Q} \sum_{m \in M} w_j \; x_{j,m} \tag{4.33}$$

Table 4.5: Sets, parameters, and decision variables Greedy solution approach

| | |
|---|---|
| ***Sets*** | |
| $M$ | set of machines $m \in M$. |
| $J$ | set of jobs $j \in J$. |
| $J'_j$ | set of successor jobs of job $j$. |
| $M_j$ | set of machines on which job $j$ can be produced. |
| ***Parameters*** | |
| $C_m$ | Latest completion time of current jobs scheduled at machine $m$. |
| $P_j$ | Total remaining processing time of job $j$ to finish the order |
| $d_j$ | Deadline of job $j$ |
| $w_j$ | Normalized weight of job $j$ in assignment model |
| ***Decision Variables*** | |
| $x_{j,m}$ | binary variable indicating job $j$ is processed in machine $m$ |

The ILP model can assign each iteration at most one job from the schedulable jobs $Q$ to each machine, this is represented in constraint (4.34). Furthermore, each job can be assigned to at most one machine, this is represented in constraint (4.35).

subject to

$$\sum_{j \in Q} x_{j,m} \leq 1, \quad \forall m \in M_j \tag{4.34}$$

$$\sum_{m \in M_j} x_{j,m} \leq 1, \qquad \forall j \in Q \tag{4.35}$$

$$x_{j,m} \in \{0,1\} \quad \forall j \in Q, m \in M_j \tag{4.36}$$

When a job is selected to be produced on a machine, the job is removed from the jobs to be scheduled and the latest completion time $C_m$ of that machine is updated. Furthermore, the latest predecessor completion is considered because it is not possible to execute a job before its predecessors are scheduled. To obtain the best set of parameters a grit search is performed with the goal of minimizing the tardiness of orders.

The model starts with an empty schedule. For each machine the latest completion time $C_m$ defines when the last job is completed at the particular machine. These are at the start initialized with 0 because no jobs are placed yet. Each iteration defines a subgroup with the jobs to schedule. A job is defined as schedulable when all its predecessors are scheduled. So the first iteration these are the jobs without predecessors.

Each iteration, the ILP model assigns jobs to a machine. For the successor jobs of the assigned job, the number of predecessors that still need to be planned is updated. Because for these jobs there is now one less predecessor yet to be produced. Furthermore, the completion time of the machine is updated. Jobs that can now be scheduled, because all its predecessors are scheduled, are added to the jobs to be scheduled. Next, the new iteration starts, and the ILP model is solved again. This process repeats until all jobs are scheduled. The pseudo code is represented in Algorithm 1.

**Algorithm 1:** Greedy scheduling Algorithm

**Result:** Schedule for all Orders in which all jobs are scheduled to a machine at a particular time point

**Initialization** An empty list of Jobs to Schedule.

job properties:list of Predecessor jobs for a job,list of successor jobs of a job, number of predecessor jobs, remaining processing time before order is completed

If a job has no predecessors the job is added to the Jobs to Schedule list.;

**while** *There are still jobs to schedule* **do**

    Solve assign ILP

    Update schedulable jobs with jobs that are now schedulable due to scheduling predecessor and drop jobs that are scheduled from schedulable jobs Update Machine completion

**end**

All jobs are scheduled, convert this to a schedule per order

### Branching

By applying branching, the search space is recursively split into smaller sub-spaces. The splitting in called branching. Branching is performed at explored nodes and is performed only when the solution of the node consists of a fractional solution. The solution is fractional when multiple schedules are selected for an order. This is possible because relaxation has been applied at the RMP and therefore $\lambda_{o,s}$ can have a fractional value.

Applying branching on the explored parent node creates two child nodes where additional constraints have been applied. The branching method is to fix the schedule with the highest fractional value. The reasoning behind this is that the MILP model selected the particular schedule because it has a positive impact on minimizing tardiness and is therefore a promising schedule. There are other possible applications that could be adopted and possibly lead to better outcomes. However, this method was chosen now and could be included in a future follow-up study.

### Node Selection

As node selection, a depth-first search is selected so that the model can arrive at an integer value relatively fast compared to breadth-first search. If an integer solution is found, this is immediately an update to the lower bound for proning the nodes yet to be explored. The depth-first search is performed until an objective is found which has a deviation within the predefined optimality gab. The disadvantage of depth-first search is that it does not guarantee to arrive at the optimal solution. In the future, this can be extended to a combination of breadth-first search, depth-first search, and best-bound search.

## 4.3 Concluding remarks

The workload optimization model is solved with a MILP model. The model respects the capacity of machines and determines target deliveries of orders, possibly by delaying the original deadlines within customer tolerance. If the production of the order is still infeasible, the model can choose to reject the order.

The scheduling model is transformed into a master problem and several sub-problems by means of Dantzig-Wolfe decomposition. The objective of the master model is to minimize the sum of tardiness of orders. The master problem chooses a schedule for each order from the set of alternative schedules of the order. By solving the master problem, the duals are generated. When the pricing problem is solved using these duals as input, a new schedule is generated for each order. If these are found to have a negative reduced cost, this schedule may have the potential to improve the objective of the master problem and is added to the set of possible schedules of the orders. This process keeps repeating until there are no orders found with negative reduced costs. However, due to the relaxation of the variables, the explored node may still consist of fractional values. This results in an unfeasible solution. By applying a branch-and-bound optimization method, the search space can then be explored to reach an integer solution. For the branching, the highest threshold selection method is applied and the search tree is explored using a dept-first search method.

Based on the given formulation, it is now clear how the production target model and the branch-and-price framework for the flexible job shop scheduling are implemented.

This concludes the methodology. Based on this chapter, it is now clear how the production target model and the branch-and-price framework are implemented. The implementation of the production target model is defined in Section 4.1. The column generation is defined in section 4.2.3 and the further implementation of the branch-and-price framework is defined in section 4.2.4. The discussed models are implemented and the results are presented in the next section.

# Chapter 5

# Results

This chapter presents the results of the proposed production target model (Section 4.1) and scheduling model (Section 4.2). The mathematical formulation and framework are explained in the previous chapter. The results of the production target model will be discussed in Section 5.3 and the results of the scheduling model will be discussed in Section 5.4. First, per model will explained how the quality is validated based on which performance measures. Next, the performance measures will be presented for the different instances.

## 5.1 Data extraction

The proposed OAS model is intended to be used on a weekly basis. To facilitate this process, the data extraction is constructed so that the data can be generated automatically using a script. The data is collected from various sources of SAP and BI. The data sources for the different properties are shown in Appendix A.

## 5.2 AME problem instances

The instances used are generated by means of AME's data system. This makes it unique compared to other papers where branch-and-price is applied in solving the scheduling problem since here a self-generated dataset is used. This makes it not only of academic value but also highly relevant to AME because they can directly apply the models to improve the production process.

An instance was exported on October 2021. At this time, all orders to be produced in the future are collected. Based on the predefined time horizon, the what-model can determine what the production target will be for the planning model. The goal is to generate a schedule for the next week to two weeks. For this purpose 5 data sets are generated with a time horizon of 5, 6, 7, 8, 9, and 10 days. Table 5.1 shows the size of the generated instances in terms of accepted orders and jobs to be scheduled by the scheduling model. The distribution of the number of jobs per order is shown in the histogram in Figure 5.1.

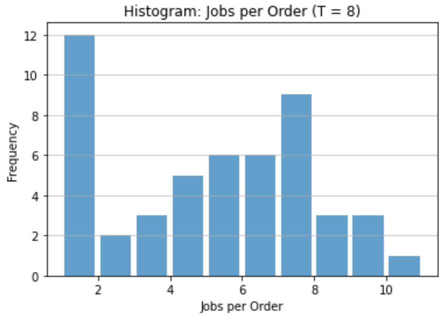



Figure 5.1: Number of jobs per order

Table 5.1: Instance sizes for different time horizons

| Time Horizon [days] | Accepted Orders | Sequential Orders | In-tree Orders | Total Jobs |
|---|---|---|---|---|
| **5** | 41 | 26 | 15 | 177 |
| **6** | 49 | 33 | 16 | 232 |
| **7** | 71 | 48 | 23 | 336 |
| **8** | 79 | 55 | 24 | 279 |
| **9** | 85 | 61 | 24 | 405 |
| **10** | 98 | 72 | 26 | 454 |

## 5.3 Workload optimization model

In this section, the results of the production target model will be evaluated. The quality of the production target model can be assessed based on the estimated utilization rates of the different machine groups and the corresponding rejected orders. This is because the model must ensure that the maximum required capacity of the machine groups is not exceeded by rejecting orders. An overview of the utilization rates of the different machine groups is shown in Figure 5.2. The number of rejected orders for these instances is shown in Figure 5.3. Both figures show different instances with a time horizon ranging from 5 to 10 days.

Figure 5.2 shows, in case of a time horizon of 5 and 6 days, there are no machine groups where the utilization rate exceeds 75 percent. Therefore, for this situation, it is expected that no orders will need to be rejected because sufficient capacity is available. Figure 5.3 confirms that for these instances no orders are rejected by the model.

In the case of the instances with a time horizon of 7, 8, 9, and 10 days, Figure 5.2 shows that the SA machine group has a utilization rate of almost 100 percent. So here SA is the bottleneck of the production process. Therefore, to ensure the SA machine group is not overloaded, orders will have to be rejected to relieve the production process. Figure 5.3 confirms that for the instance 8,9, and 10 orders are rejected.

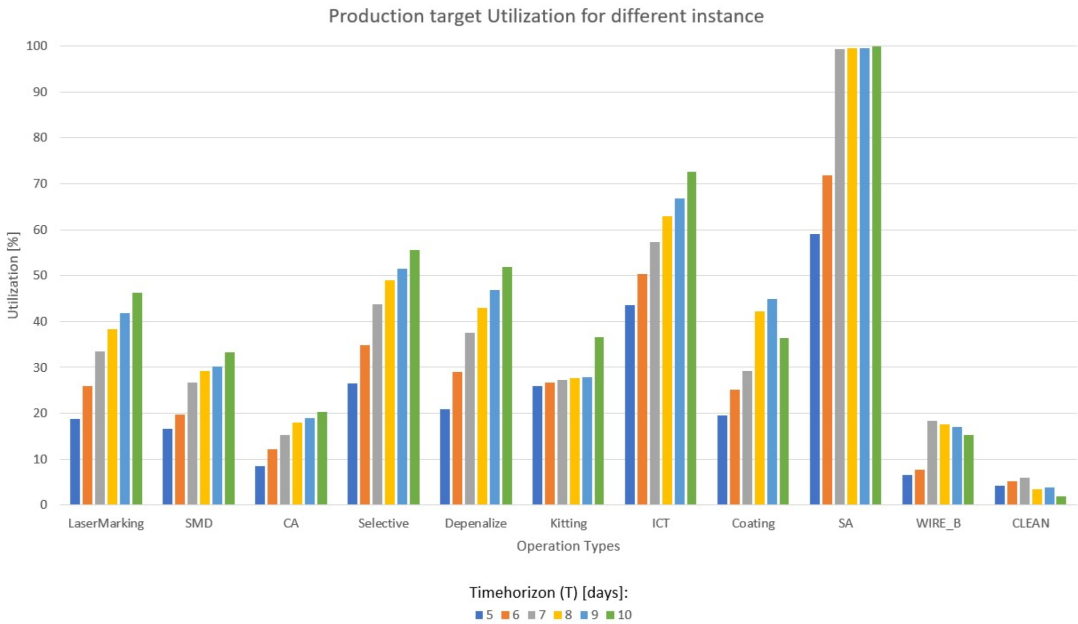



Figure 5.2: Production target utilization

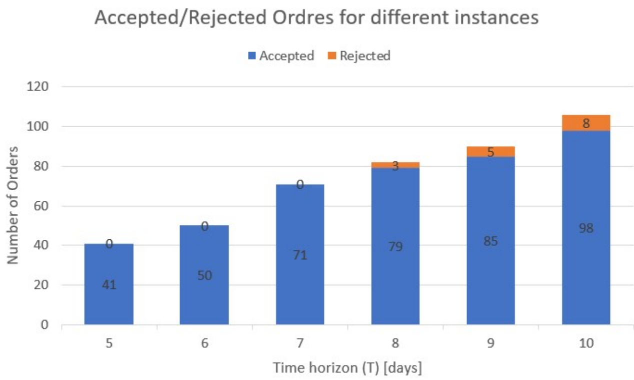



Figure 5.3: Production target Rejected Orders

Figure 5.4: Considered and rejected orders for varying customer tolerances

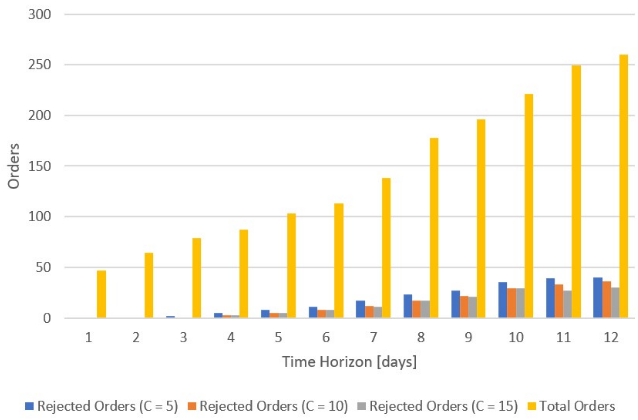
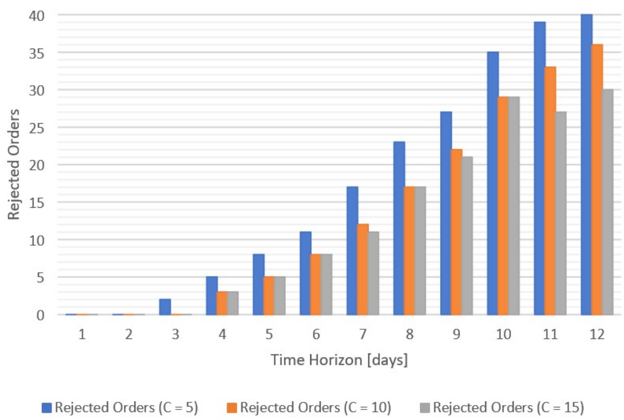



Figure 5.5: Rejected orders for varying customer tolerances




The model is examined by a sensitivity analysis where the model is tested for different customer tolerances. The customer tolerance indicates how many days the deadline of an order can be delayed before the order is rejected. It is expected that by increasing the customer tolerance, less orders need to be rejected. Figure 5.4 shows the number of accepted and rejected orders. Next, Figure 5.4 gives a more precise representation of the rejected orders for the different time windows and different customer tolerances. Here it can be seen that with an increase in customer tolerance, the number of rejected orders decreases. Therefore, it can be concluded that the model handles the customer tolerance as expected by decreasing the number of rejected orders when the customer tolerance increases.

## 5.4 Production Scheduling model

In this section, the quality of the intended production scheduling model is presented. The objective of the model is to minimize tardiness. The total tardiness of the orders and the computation time will be considered for the different instances.

Table 5.2 presents the results of the planning model. Each row presents an instance size with the predefined time granularity. Two different time granularities are presented, 60 minutes and 120 minutes. The time granularity affects the number of arcs generated for the jobs. With a time granularity of 120 minutes, fewer arcs are generated and therefore the pricing problem can be solved more efficiently. This is also reflected in the corresponding computation times. From a time horizon of 7 days, only the 120 minute granularity is considered because the calculation time would otherwise take too long. For each instance, the number of orders to be scheduled and the corresponding number of jobs are presented. Furthermore, for each instance, the initial tardiness based on the greedy scheduling approach (Section 4.2.4) is presented. Next is the BP tardiness column, which shows the final outcome of the branch-and-price framework (Section 4.2.4). Here a strong decrease can be seen in the tardiness between the outcome of the initial solution and the branch-and-price framework. Furthermore, the run time is indicated and the run time ratio indicates how much of the time the branch-and-price framework focused on calculating the master problem and the pricing problems.

Figure 5.6 compares the difference in the utilization between the production target and the production schedule generated by the scheduling model. In the case of the how-model, the available capacity in the defined time horizon was considered. The utilization is the percentage of time the machine is producing jobs. It can be seen that the utilization is slightly higher in the case of production planning. This can be explained by rounding up the process times to a multiple of the time granularity for the how-model. In the case of SA, it can be seen that the utilization is around 100%.

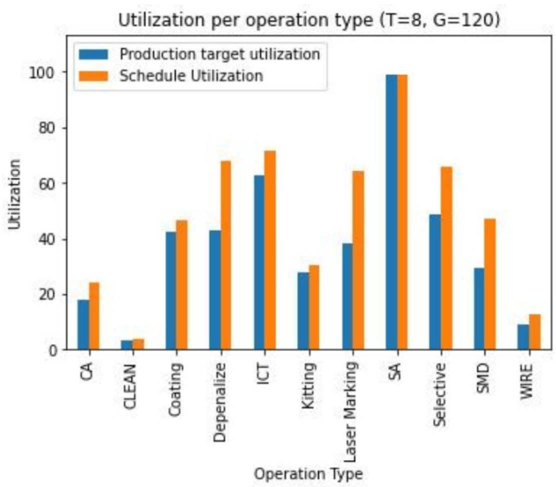



Figure 5.6: Comparison utilization Production target and schedule

Table 5.2: Results production scheduling model

| **Time Horizon** | **Time Granularity** | Accepted Orders | Total Jobs | Initial solution tardiness [min] | Root node LB tardiness [min] | B&P tardiness [min] | Comp. time [min] | Ratio Master / Runtime | Ratio Pricing / Runtime |
|---|---|---|---|---|---|---|---|---|---|
| **5** | **60** | 41 | 177 | 600 | 60 | 60 | 28m | 1% | 99% |
| | **120** | 41 | 177 | 2040 | 360 | 360 | 7m | 1.4% | 98.6% |
| **6** | **60** | 49 | 232 | 10,920 | 0 | 60 | 1h | 2.2% | 97.8% |
| | **120** | 49 | 232 | 15,240 | 300 | 480 | 22m | 2.7% | 97.3% |
| **7** | **120** | 71 | 336 | 68,880 | 4,217 | 4,560 | 13.7h | 14.3% | 85.7% |
| **8** | **120** | 79 | 378 | 144,480 | 5,184 | 6,600 | 28h | 18.2% | 81.8% |
| 9 | **120** | 85 | 405 | 222,240 | 5,206 | 6,000 | 30h | 20.4% | 79.6% |
| **10** | **120** | 98 | 448 | 222,840 | 6,340 | 7,320 | 52h | 30.8% | 69.2% |

Table 5.3: Summery Solution Quality Production scheduling (G = 120)

| Time Horizon (T) | Greedy model Objective [min] | Lower bound Root Node [min] | B&P Objective [min] | Decrease Objective | Optimality gap root node |
|---|---|---|---|---|---|
| 5 | 2,400 | 360 | 360 | 82% | 0% |
| 6 | 15,240 | 300 | 480 | 97% | 60% |
| 7 | 68,880 | 4,217 | 4,560 | 93% | 8% |
| 8 | 144,480 | 5,184 | 6,600 | 95% | 27% |
| 9 | 222,240 | 5,206 | 6,000 | 97% | 15% |
| 10 | 222,840 | 6,340 | 7,320 | 97% | 15% |

Table 5.3 shows the main results of the planning model. Two additional columns are shown. Firstly, a column indicating the percentage decrease of the greedy planning method objective and the branch-and-price objective. Secondly, the optimality gap of the objective and the lower bound of the root node.
Table 5.3 shows a large decrease in tardiness when the objective of the greedy planning model and the branch-and-price approach are compared. This implies that the column generation is well able to find better solutions. Second is the optimality gap, below 30 percent, in most instances. However, for the instance of 6, it is 60%, which indicates a large optimality gap. However, this is an optimality gap of 180 minutes overall orders, which is very little in relation to the time horizon of 6 days (8640 min) and the number of orders.

### 5.4.1 Evaluation of production scheduling

Table 2.3 shows the achieved on-time deliveries for the third quarter. To get an indication of the quality of the scheduling model, the on-time delivery was measured for the created schedules with different defined time horizons $T$. The results are shown in Figure 5.4. What is noticeable is that the percentage of intermediate On-time deliveries has improved in all work centers. It can also be seen that the desired CLIP performance, percentage of orders that are delivered on time, is achieved in all instances.

Table 5.4: On-time delivery between work centers (2020, Q3).

| **Work Center** | **On-time Delivery Measured** | **Predicted On-time Delivery** | | | | | |
|---|---|---|---|---|---|---|---|
| | | **T = 5** | **6** | **7** | **8** | **9** | **10** |
| PRD | 71% | 94% | 99% | 100% | 98% | 94% | 95% |
| WIRE | 89% | 100% | 100% | 100% | 100% | 100% | 100% |
| IM | 93% | 100% | 100% | 100% | 100% | 100% | 100% |
| CLEAN | 92% | 100% | 100% | 100% | 100% | 100% | 100% |
| SA | - | 100% | 100% | 92% | 89% | 92% | 93% |
| CLIP | | 98% | 98% | 92% | 92% | 91% | 92% |

## 5.5 Concluding Remarks

This section will summarize the results of the workload optimization model and the scheduling model

The goal of the workload optimization model is to improve and optimize the performance of the system in terms of on-time delivery through workload management and rejection of infeasible orders, in addition to ensuring optimal use of each resource. The results show the maximum utilization of resources is accomplished by rejecting orders. As a result, it can be concluded that the model is able to achieve the workload control.

The goal of the production planning model is to realize realistic production schedules where the tardiness of orders is minimized. The proposed branch-and-price framework is well able to minimize the tardiness of orders. There is a large decrease in the objective tardiness of the greedy planning method and the branch-and-price framework. This indicates that the model is capable of minimizing tardiness. The strongest argument to conclude that the model performs as intended is that the internal on-time delivery target is improved. The CLIP performance target of 85% is achieved. In the different datasets, the estimated CLIP performance ranges from 91 to 98%.

# Chapter 6

# Conclusions

This final chapter will elaborate upon the main conclusions that can be delivered from the research. Second, the limitations of the models are discussed. Finally, the recommendations will be given for both AME and recommended further research.

## 6.1 Conclusion

This section will present the conclusions of this research. This summarize the answers to the sub-questions and based on these the main question will then be answered. The research question for this master study is:

*How should the highly complex order acceptance and scheduling problem of AME be solved efficiently by using a branch-and-price search?*

The workload optimization model is solved using a MILP model. The model respects the capacity of machines and determines target deliveries of orders, possibly by delaying the original deadlines within customer tolerance. If the production of the order is still unfeasible, the model can choose to reject the order. The results show the workload control of resources is accomplished by rejecting orders. Based on the results, it can be concluded that the model is able to achieve the workload control.

The scheduling model is transformed into a master problem and several sub-problems by means of Dantzig-Wolfe decomposition. The objective of the master model is to minimize the sum of tardiness of orders. The master problem chooses a schedule for each order from the set of alternative schedules of the order. By solving the master problem, the duals are generated. When the pricing problem is solved using these duals as input, a new schedule is generated for each order. If these are found to have a negative reduced cost, this schedule may have the potential to improve the objective of the master problem and is added to the set of possible schedules of orders. This process keeps repeating until there are no orders found with negative reduced costs. If no promising columns are found, the optimal solution is found and the node explored. However, due to the relaxation of the variables, the model may still consist of fractional values. This creates an unfeasible solution. By applying a branch-and-bound optimization method, the search space can then be explored to reach an integer solution. For the branching, a highest threshold selection method is applied to define the schedule to be fixed, and for the search tree, it is explored using a dept-first search method.

Based on the results, it can be concluded that the proposed branch-and-price framework is well able to minimize the tardiness of orders. There is a large decrease in the objective tardiness of the greedy planning method and the branch-and-price framework. This indicates that the model is capable of minimizing tardiness. The strongest argument to conclude that the model performs as intended is that the internal on-time delivery target is improved. If the proposed OAS approach is used the predicted CLIP performance target of 85% is achieved. For the different datasets, the estimated CLIP performance ranges from 92 to 98%.

## 6.2 Limitations

From the academic perspective, good results are obtained regarding the quality of the model. The model is well able to solve relatively large instances. When the instance sizes are compared to other papers the model is able to solve larger instance sizes in terms of the number of orders and jobs.

For AME, the quality of the model is highly dependent on the quality of the data. Therefore, one should always strive to improve the input data. The recommendation is to improve the precision of the cycle times of job executions. AME has a decent collection of cycle times, but the execution information can be further improved in precision. Especially, recording the job executions automatically by MES will be most desired so that dynamically changing cycle times can be used by the planning models.

In work center System Assembly, there is a strong assumption by considering fixed-size teams of workers as machines. The models can be extended by defining capacity requirements of jobs as varying-size of worker teams.

In the scheduling approach, the resources are considered as machines. However, different work centers have other resources that may be binding in production. For example, System Assembly has testing equipment that is sometimes binding in job executions.

A final limitation is that the model does not take into account the availability of materials. With the current shortage of materials in Electronics manufacturing due to the temporary halt of production of materials and the growing demand for chips, orders are often postponed because they cannot be produced due to the unavailability of materials. This affects which orders can be produced and is therefore highly relevant to the planning model.

The input data set includes the availability of semi-finished components, but not the amount of available raw materials. In order to integrate the approach properly in the supply chain management of the company, raw material availability information should also be considered. Then the models can also give input to purchasing decisions.

## 6.3 Recommendations

This section will discuss the recommendations for what AME can do to further implement the intended planning method. In addition, recommendations will be given for the potential follow-up studies that presented themselves as a result of this research.

Based on the mentioned limitations, the first recommendation for AME is to dedicate more attention to the collection of the input data. The most desired way to collect input data is via the Manufacturing Execution System. This way production planning can be more autonomous by self-adapting the most recent parameters. However, a first step has already been made because the company has appointed a team to improve the quality of input data.

The second recommendation to AME is to integrate material availability into the scheduling model. This can be easily implemented in the planning model because an upper bound of the earliest possible start time is defined for each job. If it is known when the material will be available, the upper bound of the start time of a job can be adjusted accordingly. As a result, the planning becomes more accurate.

The author of this report will be involved in the business implementation of the developed scheduling methods. The business processes in the logistic department of the company will be redesigned to be able to use the developed algorithms to use the advantage of the state-of-the-art scheduling techniques. Moreover, this master thesis will provide a good base for an academic output as a journal publication.

The promising results obtained in this project made a positive effect on the collaboration of the AME company and the academy. Automated planning of manufacturing operations will be further studied as research topic of an external PhD student. All recently conducted projects, especially the current one, will provide a good base.

Further research under the academic collaboration of AME company will focus on further implementing and optimizing the model in the company. In the continued research, the effect of implementing the tool in the planning process can be further analyzed. There are, to the best of our knowledge, no papers yet where a branch-and-price framework has been implemented as a case study and so is highly relevant.

In order to further improve the scheduling method of this project, a project topic is defined as the topic of the next master thesis project. There is already a vacancy for a master thesis for the development of a prediction model that can determine the best node selection and branching strategy to explore the search area more efficiently. The goal of this master thesis is to reduce the optimality gap, improve the objective tardiness and reduce the current computation times.

# Bibliography

[1] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, P. H. Vance, Branch-and-Price: Column Generation for Solving Huge Integer Programs, Operations Research 46 (3) (1998) 316–329. doi:10.1287/opre.46.3.316. xi, 3, 19, 21, 35

[2] K. E. Stecke, Design, planning, scheduling, and control problems of flexible manufacturing systems, Annals of Operations Research 3 (1) (1985) 1–12. doi:10.1007/bf02023765. 1

[3] G. Candan, H. R. Yazgan, Genetic algorithm parameter optimisation using Taguchi method for a flexible manufacturing system scheduling problem, International Journal of Production Research 53 (3) (2014) 897–915. doi:10.1080/00207543.2014.939244. 1

[4] AME.
URL https://www.ame.nu/#about 1

[5] M. Firat, J. De Meyere, G. Tugce, L. Genga, Optimizing the workload of production units of a make-to-order manufacturing system, incorporating Surveys in Operations Research Management Science (2021). 2, 3, 17, 23, 24

[6] J. Meyere, M. Firat, Production Planning and Capacity Management of a Job Shop type Smart Manafacturing System Using Linear Optimization, Master Scriptie (2020).
URL https://pure.tue.nl/ws/portalfiles/portal/164238590/Master_Thesis_Julie_De_Meyere.pdf 13, 25

[7] J. Zhang, G. Ding, Y. Zou, S. Qin, J. Fu, Review of job shop scheduling research and its new perspectives under Industry 4.0, Journal of Intelligent Manufacturing 30 (4) (2017) 1809–1830. doi:10.1007/s10845-017-1350-2. 15, 16

[8] S. M. Johnson, Optimal two- and three-stage production schedules with setup times included, Naval Research Logistics Quarterly 1 (1) (1954) 61–68. doi:10.1002/nav.3800010110. 15, 17

[9] C. N. Potts, L. N. Van Wassenhove, A Branch and Bound Algorithm for the Total Weighted Tardiness Problem, Operations Research 33 (2) (1985) 363–377. doi:10.1287/opre.33.2.363. 16, 17

[10] R. Haupt, A survey of priority rule-based scheduling, OR Spektrum 11 (1) (1989) 3–16. doi:10.1007/bf01721162. 16, 17

[11] S. Q. Liu, E. Kozan, A hybrid shifting bottleneck procedure algorithm for the parallel-machine job-shop scheduling problem, Journal of the Operational Research Society 63 (2) (2012) 168–182. doi:10.1057/jors.2011.4. 16, 17

[12] D. E. Akyol, G. M. Bayhan, A review on evolution of production scheduling with neural networks, Computers  Industrial Engineering 53 (1) (2007) 95–122. doi:10.1016/j.cie.2007.04.006. 16, 17

[13] R. Cheng, M. Gen, Y. Tsujimura, A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies, Computers  Industrial Engineering 36 (2) (1999) 343–364. doi:10.1016/s0360-8352(99)00136-9. 16, 17

[14] R.-H. Huang, T.-H. Yu, An effective ant colony optimization algorithm for multi-objective job-shop scheduling with equal-size lot-splitting, Applied Soft Computing 57 (2017) 642–656. doi:10.1016/j.asoc.2017.04.062. 16, 17

[15] M. Nouiri, A. Bekrar, A. Jemai, S. Niar, A. C. Ammari, An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem, Journal of Intelligent Manufacturing 29 (3) (2015) 603–615. doi:10.1007/s10845-015-1039-3. 16, 17

[16] J. M. van den Akker, J. A. Hoogeveen, S. L. van de Velde, Parallel Machine Scheduling by Column Generation, Operations Research 47 (6) (1999) 862–872. doi:10.1287/opre.47.6.862. 16

[17] J. M. van den Akker, J. A. Hoogeveen, J. W. van Kempen, Parallel Machine Scheduling Through Column Generation: Minimax Objective Functions, Lecture Notes in Computer Science (2006) 648–659doi:10.1007/11841036_58. 16, 17

[18] H. M. Wagner, An integer linear-programming model for machine scheduling, Naval Research Logistics Quarterly 6 (2) (1959) 131–140. doi:10.1002/nav.3800060205. 17

[19] Z. A. Lomnicki, A “Branch-and-Bound” Algorithm for the Exact Solution of the Three-Machine Scheduling Problem, Journal of the Operational Research Society 16 (1) (1965) 89–100. doi:10.1057/jors.1965.7. 17

[20] A. S. Manne, On the Job-Shop Scheduling Problem, Operations Research 8 (2) (1960) 219–223. doi:10.1287/opre.8.2.219. 17

[21] G. Desaulniers, J. Desrosiers, M. M. Solomon, Column generation, Vol. 5, Springer Science & Business Media, 2006. 17, 19

[22] S. Mestry, P. Damodaran, C.-S. Chen, A branch and price solution approach for order acceptance and capacity planning in make-to-order operations, European Journal of Operational Research 211 (3) (2011) 480–495. doi:10.1016/j.ejor.2011.01.002. 17, 18

[23] M. Nawaz, E. E. Enscore, I. Ham, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, Omega 11 (1) (1983) 91–95. doi:10.1016/0305-0483(83)90088-9. 17

[24] J. Adams, E. Balas, D. Zawack, The Shifting Bottleneck Procedure for Job Shop Scheduling, Management Science 34 (3) (1988) 391–401. doi:10.1287/mnsc.34.3.391. 17

[25] W. Xia, Z. Wu, An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, Computers  Industrial Engineering 48 (2) (2005) 409–425. doi:10.1016/j.cie.2005.01.018. 17

[26] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers  Operations Research 13 (5) (1986) 533–549. doi:10.1016/0305-0548(86)90048-1. 17

[27] S. Łukasik, S. Żak, Firefly Algorithm for Continuous Constrained Optimization Tasks, Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems (2009) 97–106doi:10.1007/978-3-642-04441-0_8. 17

[28] S. Voorn, Coordinated production scheduling for a High-Mix Make-to-Order manufacturing company, Master thesis 2021 (2021).
URL https://pure.tue.nl/ws/portalfiles/portal/174234289/Master_Thesis_Suzanne_Voorn.pdf 17

[29] S. A. Slotnick, Order acceptance and scheduling: A taxonomy and review, European Journal of Operational Research 212 (1) (2011) 1–11. doi:10.1016/j.ejor.2010.09.042. 18

[30] X. Li, J. A. Ventura, K. A. Bunn, A joint order acceptance and scheduling problem with earliness and tardiness penalties considering overtime, Journal of Scheduling 24 (1) (2020) 49–68. doi:10.1007/s10951-020-00672-5. 18

[31] S. Wang, J. Wan, D. Zhang, D. Li, C. Zhang, Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination, Computer Networks 101 (2016) 158–168. doi:10.1016/j.comnet.2015.12.017. 18

[32] E. Hans, Resource Loading by Branch-and-price Techniques: Models and Algorithms, Beta, Research School for Operations Management and Logistics, 2002. 18

[33] B. Maenhout, M. Vanhoucke, Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem, Journal of Scheduling 13 (1) (2009) 77–93. doi:10.1007/s10951-009-0108-x. 19

[34] Y. Pochet, L. Wolsey, Production Planning by Mixed Integer Programming, 2006th Edition, Springer Publishing, New York, Verenigde Staten, 2006. 20

[35] Bilge, M. Fırat, E. Albey, A parametric fuzzy logic approach to dynamic part routing under full routing flexibility, Computers Industrial Engineering 55 (1) (2008) 15–33. doi:10.1016/j.cie.2007.11.013. 36

# Appendix A

# Data collection

The data was collected from various sources from SAP and BI. The data sources of the required properties are shown in Table A.1.

Table A.1: Data Sources

| Work Center | Information | Data File | Data Source |
|---|---|---|---|
| All | Stock info | Stock Report | SAP |
| All | Assumptions | Assumption file | Info collected from AME personnel |
| All | Sub products necessary for production | Bill of Materials | SAP |
| All | Scrap information | Production Quality Report | SAP BI |
| All | Production Orders | Delivery Schedule | SAP BI |
| All | Forecast Products | Forecast Levels | SAP |
| All | Batch Sizes | Historical Production Data | SAP |
| IM | Setup & production time, machine eligibility | IM planning file | Orion |
| PRD | Operations needed per PN | PAD Documents | Orion |
| PRD | Scrap info | Production Quality Report | SAP BI |
| PRD | PCB info | PCB Report | SAP |
| PRD | SMD cycle times | SMD planning file<br>Fuji cycle time report | AME server<br>SAP |
| PRD | SMD setup times | Fuji uptime report | SAP |
| PRD | Selective setup times | Selective uptime report | SAP |
| PRD | Selective cycle times | Selective cycle time report | SAP |
| PRD | Conventional Assembly target | PRD assembly targets file | AME server |
| PRD | ICT times | ICT targets file | AME server |
| SA & CLEAN & WIRE | Processing time | Approximated processing times | Financial Data Analysis |