NTNU
Norwegian University of
Science and Technology

Master's thesis

2019

Haakon Alexander Aarholt

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Haakon Alexander Aarholt

Genetic Algorithm for Job Shop Scheduling in High-Variety, Low-Volume Production Environments

June 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Genetic Algorithm for Job Shop Scheduling in High-Variety, Low-Volume Production Environments

Haakon Alexander Aarholt

Engineering and ICT
Submission date: June 2019
Supervisor: Anita Romsdal
Co-supervisor: Erlend Alfnes

Norwegian University of Science and Technology
Department of Mechanical and Industrial Engineering

Preface

This Master's thesis concludes my Master of Science in Engineering and ICT at the Department of Mechanical and Industrial Engineering.

I would like to thank Anita Romsdal for her extensive involvement and guidance for this Master's thesis with her valuable knowledge on advanced planning and scheduling systems and scientific writing.

I also want to thank Erlend Alfnes for helping me organizing the case study in addition to getting me in touch with case company data. Special gratitude also goes to Hans-Henrik Hvolby for his knowledge within the fields of advanced planning and scheduling systems, and for establishing assumptions for the development of the optimization model. Gratitude also goes to Andreas Dypvik Landmark for his feedback on the thesis towards the end. Lastly, I want to thank for the love and support which my family, my friends and my girlfriend have given me.

Abstract

Production planning and scheduling in high-variety, low-volume production environments is a challenging task, in times where short lead times and highly customized products are sought after. The act of scheduling manufacturing processes becomes increasingly computationally intractable when also the number of unique production routings increases, often resulting in sub-optimal production plans being produced. Advanced systems for coping with the aforementioned challenges do exist, but yet production planners tend to hold on to traditional planning approaches as becoming familiar with new systems is regarded as a hassle, and the set of advantages often are looked past.

This study aims to identify characteristics in jobbing environments belonging to the field of the engineer-to-order production process and to develop three variants of an optimization model which aim to obtain optimal schedules based on different objective functions. The results from the optimization models are in agreement with the literature, depicting exceptional abilities to solve large optimization problems. Furthermore, effective capabilities of advanced planning and scheduling (APS) systems are identified, and parallels are drawn to the production planning and scheduling capabilities which become enabled by the optimization models.

The results depict optimized schedules based on the input data and constraints, according to various objective functions. Schedules are produced with minimized total makespan, satisfied deadlines, with a late start approach, and a combination of the three aforementioned traits. Promising results are obtained by setting accurate constraints enabled by the implementation of an APS system, integrated among supporting systems. In turn, rescheduling capabilities are enabled, rendering dynamic production planning and scheduling capabilities that are able to respond to disruptions on the shop floor.

Sammendrag

Produksjonsplanlegging i produksjonsmiljø som er preget av store produktvarianter og et lavt volum er en stor utfordring i tider hvor korte ledetider og svært tilpassede produkter er høyt ettertraktet. Arbeidet med å planlegge produksjonsprosesser er svært omfattende, og blir beregningsmessig uhåndterlig når antall unike produksjonsruter økes, og resulterer ofte i at suboptimale produksjonsplaner blir laget. Avanserte systemer for å takle de ovennevnte utfordringene finnes, men produksjonsplanleggere har en tendens til å holde fast på tradisjonelle planleggingsmetoder ettersom strevet med å bli kjent med nye systemer blir sett på som et problem, og fordelene en slik implementasjon medfører ofte blir forbisett.

Denne studien tar sikte på å identifisere karakteristikker i produksjonsmiljøer hvor flere jobber utføres på en og samme tid, på et utvalg av maskiner, og utvikle en optimeringsmodell som tar sikte på produsere optimale tidsplaner basert på ulike objektivfunksjoner. Resultatene fra optimaliseringsmodellene ligner de fra litteraturen, og viser gode evner til å løse store optimeringsproblemer. Videre identifiseres styrker ved avanserte planleggingsystemer, og paralleller trekkes fra slike systemer til de utviklede modellene.

De konkrete resultatene skildrer optimaliserte tidsplaner basert på data og begrensninger som modellen håndterer i henhold til ulike objektivfunksjoner. Tidsplaner er produsert med minimert gjennomløpstid, tidsfrister som er holdt, med sen-start tilnærming, og en kombinasjon av de tre nevnte trekkene. Lovende resultater oppnås ved å etablere nøyaktige begrensninger, muliggjort ved implementering av et avansert system for planlegging, integrert blant støttesystemer. Dette muligjør effektive omplaneringsfunksjoner som gir dynamisk produksjonsplanlegging og planleggingskapasitet, i stand til å reagere raskt til forstyrrelser på verkstedsgulvet.

Table of Contents

List of Tables

List of Figures

Abbreviations

ETO	=	Engineer-To-Order
MTO	=	Make-To-Order
ATO	=	Assemble-To-Order
MTS	=	Make-To-Stock
HVLV	=	High-Variety, Low-Volume
JSSP	=	Job Shop Scheduling Problem
APS	=	Advanced Planning and Scheduling
ERP	=	Enterprise Resource Planning
MRP	=	Material Requirements Planning
MRP II	=	Manufacturing Resource Planning
DIFOT	=	Deliverey In Full, On Time
CODP	=	Customer Order Decoupling Point

Chapter 1

Introduction

1.1 Background and motivation

”In order to achieve and maintain a competitive edge in the world marketplace, manufacturing companies must produce high-quality products at low cost with increasing variety, over shorter lead times”, (Ghalayini et al., 1997). Manufacturing companies in various production environments are characterized by the aforementioned requirements, and one production environment is greatly affected by the requirements mentioned above. Engineer-to-order (ETO) production environments are dynamic, and the planning and control functionality is affected by the production of highly customized products, (Sriram et al., 2012). Engineer-to-order processes belong to the production environment known as high-variety, low-volume, further referred to as HVLV. (Buetfering et al., 2016). The environment is characterized by complex and non-repetitive production routings, long lead times, causing production planning and scheduling to be a challenging task and thus creating difficulties in accurately estimating the completion date. Job shops, a manufacturing process belonging to the field of ETO environments, is characterized as operating with numerous jobs, each consisting of operations to be completed on a selection of machines in various sequences. The major challenge for this type of planning is that the corresponding optimization problem, namely the aim of placing operations in such a sequence that a close-to-optimal schedule based on objectives is obtained, becomes computationally intractable as the number of products, markets, and production sites increases in the supply chain network, (Shah and Ierapetritou, 2012).

1.2 Problem formulation

Production planners in HVLV environments lack the ability to analyze real-time production capabilities, and the ability to rapidly and frequently adapt to incidents that arise in the manufacturing environment. Establishing accurate constraints for all processes from the time a customer order is accepted until the order is delivered is a challenging task. Moreover, the optimization problem of scheduling multiple jobs’ operations on various machines is computationally challenging. These tasks have been avoided by utilizing similar executed plans in order to establish a new production plan and schedule. However, planning in such a way provides little to no seamless flow of information and does not assist in obtaining a centralized data sharing environment, thus resulting in independent and static company departments unable to cope with the dynamic environment, (Stadtler, 2005).

Advanced planning and scheduling (APS) systems offer functionality which assists with solving the aforementioned challenges by enabling frequent rescheduling, real-time access to external data, in addition to a seamless integration of plans (Ivert, 2012). These abilities depict functionality, which is less existent in more traditional planning systems such as Microsoft Excel and enterprise resource planning (ERP) systems. Studies regarding successful integrations of APS systems in manufacturing companies have been done, (Wiers, 2002), (Zoryk-Schalla et al., 2004), (Stadtler and Kilger, 2002), but companies are still reluctant to invest in them due to challenges and costs of which an implementation often entails. This study aims to shed light on the benefits of effective scheduling, benefits which may be provided by the APS systems' production planning and scheduling module, in addition to further advantages provided by the remaining modules of APS, described in Section 3.3.4.

The main focus of this study is the APS systems' core optimization model, which handles production planning and scheduling. A genetic algorithm, inspired by the process of natural selection, is proposed, aiming to optimize the production planning and scheduling based on an objective function, containing parameters aiming to maximize or minimize specific values. The objective function of the final developed model aims to minimize the number of jobs which are delayed in a given set of orders, in addition to minimizing the elapsed time from the start of the first task, to the last task, further referred to as the makespan. By realizing this objective, users of these models have the ability to continually perform rescheduling due to disruptions on the shop floor, evaluate future customer orders, and thus assess and increase the ability to deliver in full, on time, (DIFOT), a fundamental measurement depicting a supply chain's delivery performance, (Boonsothonsatit, 2016).

1.3 Research objectives

The primary objective of this thesis is to develop an effective, efficient and generic optimization model, based on a genetic algorithm, that performs job shop scheduling in regards to minimizing the plan's total makespan and considering the orders' delivery dates. Furthermore, the model is to depict benefits which manufacturing companies may draw from by investing in an APS system.

Three objectives are formulated to achieve the main objective.

1. Map and analyze characteristics of HVLV production environments and challenges of job shop scheduling

2. Utilize an effective genetic algorithm and adapt the algorithm to handle job shop scheduling with characteristics of HVLV environments

3. Compare the developed optimization models to an APS system's core model

The research objectives form the basis for the discussion in Chapter 7, and the fulfillment of the objectives is evaluated in Chapter 8.

1.4 Scope

The main focus of this thesis is the development of an optimization model based on a genetic algorithm for effectively scheduling the operations in job shops, characterized as revolving around jobs consisting of operations to be processed in a specific sequence on a selection of machines. The models are limited to the operational planning of industries operating with the scheduling of jobs, and the research is directed towards manufacturing industries operating in an HVLV environment. Focus has also been on identifying the current operational production planning practices among manufacturers in such an environment in order to develop models to tackle the documented challenges, elaborated in Section 5.5.1. Specifically, the operational production planning at a case company has been utilized as the basis for the research conducted and presented in this study. Assumptions have been made, found in Section 3.2.1, in order to form the basis for the development of the models, in addition to simplifying modeling aspects which would hinder the running of the models. The models do not consider uncertainties in the supply chain, such as demand uncertainty, supply uncertainty, breakdowns of machines or fluctuating inventory holding, but is instead based on deterministic values for the input. A standard generic structure for APS systems has been identified and will be used to display advantages and challenges with those type of systems.

1.5 Contributions

This study contributes to depicting the strengths of APS systems' core by developing a scheduling algorithm in the context of APS systems. The models are based on a paper which optimizes jobs' operations solely based on the makespan. An additional contribution of this paper is further developing the algorithm to consider jobs' deadline. Furthermore, the ability to frequently reschedule is granted. The results in Chapter 6 show schedules with multiple jobs which satisfy strict deadlines and depict effective rescheduling capabilities.

1.6 Structure

The remainder of this thesis is structured as follows. Chapter 2 presents the research methodology, consisting of a literature study, an operations research methodology, and a case study. Furthermore, Chapter 3 presents the theoretical background, consisting of findings from the literature study, which is relevant for the development of the optimization models. Chapter 4 presents assumptions, characteristics, and input parameters to the developed optimization models. Furthermore, Chapter 5 describes the case company, granting industry data to further develop the optimization models and test real data. Chapter 6 depicts the obtained results from the developed optimization models. Chapter 7 analyzes the aforementioned results and argues the advantages and disadvantages of the various approaches. Finally, conclusions are drawn from the results and the discussion in Chapter 8.

Chapter 2

Methodology

Methodology is the theoretical, systematic reasoning of methods applied to a field of study. Typically, it includes concepts such as quantitative or qualitative techniques, aiming to underline which set of methods can be best applied to a research area, (L BERG, 2001). Quantitative approaches make use of mathematical tools to analyze data, while qualitative approaches are not as concerned regarding counts or measures, but rather researching why and how questions by use of constructivism, and interpretation, (Croom, 2010).

This master's thesis is carried out with a theoretical and an empirical part, utilizing two main research methods: a case study and a literature study. The research presented is based on a descriptive and prescriptive approach, thus identifying the AS-IS situation in the industry and challenges that may entail, followed by proposing new methods to solve the documented challenges. The descriptive approach of identifying common practices and challenges is regarded as a qualitative approach while developing the optimization models to propose new methods to solve challenges is regarded as a quantitative approach.

The study is carried out in accordance to a methodology for operations research (OR) displayed in Table 2.1, adapted from (Hillier, 2010), highlighting a selection of steps relevant for meeting the objectives of this thesis. The methodology consists of six steps, of which the four highlighted are to be carried through in this thesis. The first step is to define the problem and collect data relevant to it. Next comes the formulation of a mathematical problem. The following step is deriving solutions from the developed models and interpreting them. Furthermore, testing of the models ensues and iteratively refining them to obtain desired results. The last two steps encompass implementation steps, regarded as tasks not suitable for the time frame of this thesis, but which are relevant for further work, and will therefore be discussed thereafter.

Step	Described steps of the OR methodology
1	Define the problem of interest and gather data
2	Formulate a mathematical model to represent the problem
3	Derive solutions from the model
4	Test the model and refine it as needed
5	Prepare for changes requested by management
6	Implement the system

Table 2.1: Operations research methodology, adapted from (Hillier, 2010). The four highlighted steps are to be carried out in this thesis

2.1 Literature study

Researching relevant literature is an essential feature of any academic project. A literature study assists in identifying the research that has been done on the topics and recent advancements that have been made to arrive at the topics' state-of-the-art. There is great value in placing the research in perspective and discovering findings which may be used to support a paper's resulting findings, and uncovering areas where research gaps exist, (Webster and Watson, 2002).

The collection of data enabled the initiation of tackling the first objective described in Chapter 1.3. Further data collection required to introduce the theoretical topics in Chapter 3 was obtained by performing systematic searches with relevant keywords, depicted in Table 2.2, using different search engines. The search engines Google Scholar and Oria were frequently used among each other to identify relevant papers. The literature assisted in identifying approaches to similar problems, gaps in the literature, in addition to shedding light on the addressed problem.

The literature searches shed light on four theoretical fields. The first field of interest was the environment of which the industries of interest operate in, namely with high-variety, low-volume (HVLV) manufacturing. (Adrodegari et al., 2015) and (Stavrulaki and Davis, 2010) proposed several characteristics of the HVLV environment. Identifying characteristics of this environment was deemed essential in order to proceed with the following field of interest, namely production planning and scheduling. The literature revealed common approaches utilized in the past, approaches no longer feasible, and approaches regarded as promising for the future. The main source of literature to depict effective scheduling approaches was (Yin et al., 2011), with assumptions and logic corresponding to the problem formulation. The third field of interest was advanced planning and scheduling (APS) systems, argued in Chapter 3.3 to be the state-of-the-art type of system for manufacturing companies in order to stay competitive, with planning and scheduling functionality enabled by an optimization model. As APS systems have been around for longer than two decades, there is extensive literature to be researched. Research papers such as (Ivert, 2012), (Stadtler and Kilger, 2002), (Umble et al., 2003) have been thoroughly studied to grasp the essence of APS systems, and (Meyr et al., 2008) has provided in-depth knowledge to the structure of APS systems. Lastly, artificial intelligence, with a focus on genetic optimization algorithms, was an essential field to analyze in order to develop the desired mathematical models. Essential literature for developing the initial models of which the results in Section 6.1 were obtained was (Yin et al., 2011) and (Karaboga and Basturk, 2008).

As the investigated fields have been extensively researched, the number of articles obtained through literature searches were numerous. Grasping the essence through skim-reading the abstract and the conclusion proved to be beneficial in identifying essential papers. When the paper was deemed relevant, a more thorough analysis was performed.

In order to obtain relevant articles to provide information regarding the fields of interest, several keywords were frequently utilized in search engines, depicted in Table 2.2.

Table 2.2: Keyword searches

Keyword set 1	Keyword set 2
High-variety, Low-volume (HVLV) Engineer-to-order (ETO) Make-to-order (MTO) Operations Research (OR) Manufacturing	Definition Environment Production planning
Scheduling	Job shop scheduling (JSS) Multi-objective scheduling
Artificial intelligence Optimization algorithms	Evolutionary algorithms Genetic algorithms
Advanced planning & scheduling (APS) Enterprise Resource Planning (ERP)	Optimization model Scheduling system Software modules

The findings of the literature study are presented in Chapter 3.

2.2 Model development

The study is carried out in accordance to the methodology for operations research displayed in Table 2.1, consisting of a total of six steps. The four steps carried out in this thesis are described in the following sections.

2.2.1 Define the problem of interest and gather data

The first step of the operations research methodology is described as defining the problem of interest and the gathering of data. The problem of interest is proposed in Section 1.2, sparked by literature and by the case company. The problem in existence was verified through meetings, assisting with company data depicting the problem. The distribution of data, however, became available at a late stage, rendering the specific company problem and their data structure unknown and in need for speculation until eventually obtained. Researching the fields relevant for developing the models were essential, enabled by using literature searches in order to grasp an understanding of the state-of-the-art, a method presented in Section 2.1.

2.2.2 Formulate a mathematical model

”A mathematical model is a representation in mathematical terms of the behavior of real devices and objects”, (Dym and Ivey, 1980). Figure 2.1 depicts a real and a conceptual world. The real world consists of phenomena, the source for observations being made. An attempt is made to accurately mimic the nature of the real world, identified as observations, then perform analyses using models to obtain results, identified as a prediction of the outcome in the real world. In the context of operations research, the mathematical model’s core concept is its objective function, defined by decision variables, weighted accorded to desired objectives. The objective function’s motive is to maximize or minimize a value, to the best degree possible.

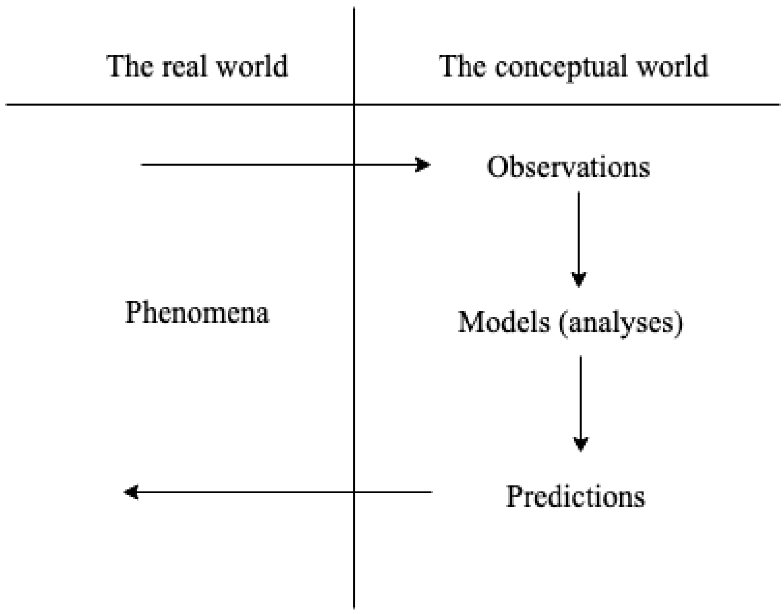

Figure 2.1: Conceptual models related to observation made within the real world, adopted from (Dym, 2004)

In order to mimic the nature of the real world in a mathematical model, establishing assumptions is necessary to define the problem scope. The combination of all assumptions forms the basis for the mathematical models. Influencing parameters may require to be simplified, either regarding which parameters to include, or converting uncertain and varying values to constants. The first task was to implement the artificial bee colony algorithm and obtain solutions to well-researched problems. After that, the model required adaption in order to accept company data. Company data had to be structured such that the model would recognize the input, and some values tend to be problematic to determine, such as the breakdown of machines, the occurrence of rush orders, machines’ changeover-times, or incidents of products not meeting the required quality. These values were therefore not included in the algorithms, rendering the developed models more idealized than in reality. Finally, a further developed model was obtained to handle complex job scheduling with deadlines considered.

There exists an extensive range of optimization approaches. Some models settle with the first feasible solution, while other models aim to approach the optimized solution to a problem. Succeeding in obtaining a close-to optimized solution tends to be a challenging task, a task consisting of defining the objective function and implementing operators to traverse the vast number of possible solutions. The initial step to formulating the mathematical model is accurately defining the objective function by communicating with stakeholders in order to obtain information regarding desired objectives. Furthermore, establishing constraints which resemble the real world is essential, or else the production schedule will be unfeasible. Input data and model specifications are further described in Chapter 4.

2.2.3 Derive solutions from the model

The solutions were obtained using a MacBook Pro with a 2.9 GHz Intel Core i7 processor and 16 GB RAM. The optimization models were developed in the Java programming language, and the figures were generated with the Python programming language. Input files with data were entered directly into the program. The development of the model is further described in Chapter 4. The implementation of the models and the obtained results appeal to research objective 2 in Section 1.3.

2.2.4 Test the model and refine it as needed

As soon as the step of deriving solutions from the model is completed, additional testing is necessary to obtain results which are desired. Stakeholders may request new functionality which becomes implemented by performing iterative modeling. Analyzing the results and communicating with stakeholders is essential in order to aim for more optimized solutions, derived by iterative modeling, (Jonsson and Backstrom, 1995). Refining of the models was performed according to feedback from meetings where the optimization models were presented.

2.3 Case study

Case studies, as one of the first types of research used in the field of qualitative methodology, have accounted for a large proportion of research presented in scientific literature, (Starman, 2013). Although case studies often have been regarded as part of a qualitative methodology, it may also encompass quantitative research and a combination of the two. (Sagadin, 2004) argues that ”qualitative and quantitative results should complement each other to create a meaningful whole according to the object and purpose of the investigation.” This thesis is carried out with the aforementioned approach, utilizing the case study as a source to numerical data and in-depth analysis of the industry.

Information regarding the case company has been obtained by utilizing several sources. Data and characteristics of the case company have been collected through meetings, from written project documentation, previous project- and master theses, in addition to research papers where the case company has been involved or mentioned. Data regarding four company products have been adapted to depict results from one of the optimization models. Specific input data of these products were obtained by analyzing an Excel sheet provided through meetings. Further descriptions of the case company and the data are presented in Chapter 4 and Chapter 5.

Chapter 3

Theoretical background

As the research area of this study comprises of high-variety, low-volume manufacturing, and assumptions for the models have been based on characteristics of this environment, an elaboration of the environment is necessary. Production planning and scheduling differ significantly among environments, therefore obtaining an understanding of the environment motivates for the benefits of effective scheduling and argues for the assumptions made when developing the models. In addition, scheduling theory, with a focus on characteristics of the job shop scheduling problem, is highly relevant for the engineer-to-order industry, the case company, in addition to the development of the models. Moreover, planning systems have been identified and compared with each other in order to propose the state-of-the-art within assisting the relevant production environment. Furthermore, as the models which have been developed are based on research done in the field of artificial intelligence, (AI), utilizing an algorithm belonging to the group of genetic algorithms, an introduction to the field will be beneficial in order to follow the logic of the genetic algorithm.

3.1 High-variety, low-volume manufacturing

The term itself, HVLV, describes the production of a large variety of products, while the quantity of products is low. Accurate establishment of a low volume quantity is regarded as a difficult task as a specific volume for one industry sector differs from another, (Amaro et al., 1999). The definition of low volume for this study has been set to less than 400 products produced annually, to correspond with the case company. The term HVLV is generally linked to manufacturers operating with engineer-to-order or make-to-order processes, (Portioli-Staudacher and Tantardini, 2012).

Engineer-to-order (ETO) is a manufacturing process where the product is engineered to meet specifications requested by the placed order. All steps; production, procurement, the final assembly, and delivery are managed in response to the customer order. In turn, this entails a wide range of customized requests, a high level of customer interaction, and difficulty in predicting costs for the entire process, (Mbaskool, 2019).

An additional manufacturing process which encompasses HVLV products is **make-to-order** (MTO), which possesses similar traits as the ETO process, with some distinctions. Product design is generally completed prior to the customer order being placed, leaving manufacturing and assembly as the following tasks to be completed before delivery takes place. Units produced by an MTO process are generally manufactured using standard components, while in time also using customized parts.

Although ETO and MTO share similar traits, the challenges in MTO are more prevalent in ETO. The customer involvement takes place at an earlier stage in ETO, entailing longer delivery lead times as the high degree of customization naturally requires more time than if the task is not present. The diverse jobs to be processed, and thus variations in processing times and routings result in challenging production planning and control. The additional steps a manufacturing process consists of, combined with the lack of repetitiveness, the greater becomes the challenge to predict the production time.

A trade-off analysis as to which key performance indicators to value the most has to be done for each planning period. Increased inventory holding and overtime working contribute to meeting the delivery dates and thus increasing customer satisfaction, but to what cost? Coping with an increasingly competitive market may be assisted with more stable, efficient, and visible planning and production processes. Adapting to unforeseen incidents causing eruptions in production plans by rescheduling is undoubted of great value, and can only be carried out by enabling rescheduling of the production plan at short notice. Reducing the delivery lead time, more importantly, not exceeding the delivery date, is argued to significantly affect the competitiveness in high-variety, low-volume (HVLV) manufacturing environments, (Adrodegari et al., 2015).

Project and job-shop type processes tend to dominate the ETO and MTO supply chains in order to ensure production flexibility, (Stavrulaki and Davis, 2010). The two processes aim to handle customized orders, with the former type more dedicated to complete an order in its entirety, while the latter type typically handles a more extensive range of items at the same time. A further distinction among the two follows.

Jobbing processes take place in job shops, which are further described in Section 3.2.1, and are characterized by enabling the manufacturing of small quantities of a great variety of products, each of which is custom designed and therefore requires a unique sequence of processing steps. By not limiting oneself to a particular type of product, the client base may be increased by remaining flexible in the variety of products produced. The waiting time for accessing a specific piece of equipment differs, but it may be extensive as some machines become overloaded, while other machines may remain idle over an extended amount of time.

Project processes deal with discrete, highly customized products. The timescale of these products tends to be in the range of months or even years. Large parts of the activities involved in the manufacturing of the product may be ill-defined and uncertain due to continuous customer interaction, such as in projects run by shipbuilding and construction companies. Each product occupies such a large part of the capacity in such degree that the manufacturer's resources are more or less devoted exclusively to it, (Slack et al., 2010).

Characteristics of project processes and jobbing processes are depicted in Figure 3.1, among other processes possessing different characteristics regarding variety, volume, and repetitiveness, (Hayes and Wheelwright, 1979). The type of product mix described in jobbing and project processing is consistent with the description of HVLV above and will be the processes of interest throughout the thesis.

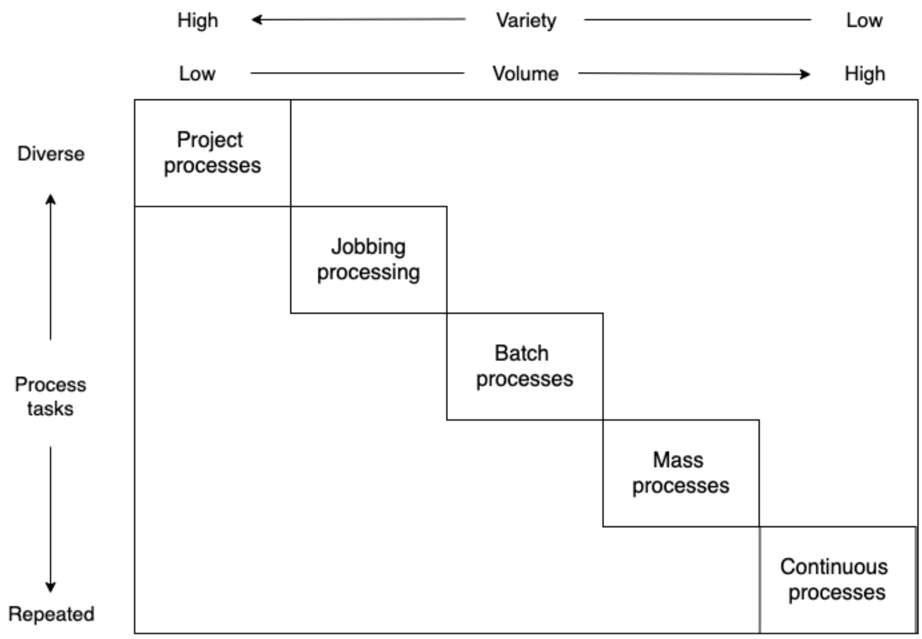

Figure 3.1: Variety and volume matrix for products, adapted from (Hayes and Wheelwright, 1979)

The different types of processes depicted in Figure 3.1 entail a different manufacturing strategy, with the act of scheduling largely dependant on the type of products to be manufactured. Similarly, the specific point in the value chain at which the customer influences the production activities, referred to as the customer order decoupling point (CODP), varies among the various production approaches. Figure 3.2 depicts the CODP of the four most common production approaches, displaying ETO as the approach of which the customer first triggers the production activity, in the engineering phase.

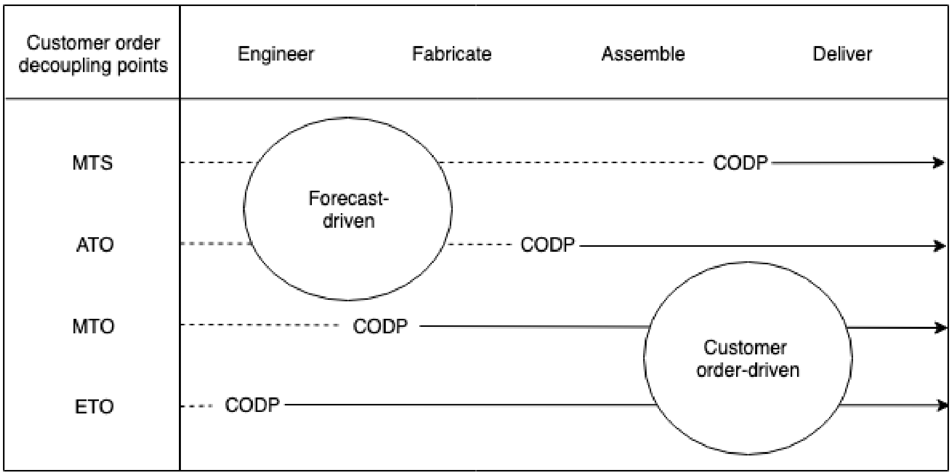

Figure 3.2: Customer order decoupling point for the various production approaches, adapted from (Sharman, 1984)

Scheduling is a problem that arises in a vast number of fields, consisting of the efficient allocation of resources to perform tasks in order, concerning time. In manufacturing, a well-researched scheduling problem is defined as the job shop scheduling problem, with tasks that require processing on a set of machines, in a specified order, (Dumitrescu et al., 2007). The process of prioritizing one job over another is based on the schedule's objective.

3.2 Production planning and scheduling

Scheduling in an effective manner assists the manufacturing world in meeting their performance criteria. Key performance indicators such as manufacturing lead times, meeting due dates, machine utilization, inventory costs, and customer satisfaction are all greatly influenced by the degree of efficiency which the schedule provides, (Tamilarasi et al., 2010). In order to meet objectives and stay competitive, developing effective scheduling approaches becomes increasingly essential. The costs related to delayed projects are not only economical but may also be hidden, although just as punitive. Customer satisfaction levels are greatly affected, and although customer satisfaction does not guarantee repurchases, it still plays an important role in ensuring customer loyalty and retention, (Singh, 2006).

Multi-operation multi-machine scheduling is an extensive research area due to aspects that arise in production planning, and computer control, (Mao, 1995). The complexity of combinatorics is the main source of struggles met in this environment, namely the number of possible solutions quickly getting out of hand when operating with more than three machines. (Röck, 1984). Three common types of shop scheduling exist, as described in Table 3.1.

Table 3.1: Different types of shop scheduling

Term	Description
Open shop scheduling	The order in which a job passes through the shops is immaterial
Flow shop scheduling	All jobs have the same shop ordering
Job shop scheduling	All jobs may have different shop orderings

The type of scheduling relevant for this study is the job shop scheduling, as jobs to be scheduled may have different steps that are required for manufacturing the given parts. There exist several types of manufacturing environments; however, as introduced in Section 3.1, the field of interest is jobbing processes, which take place in job shops. An introduction to job shop scheduling, and its corresponding and well-researched problem, follows.

3.2.1 Job shop scheduling

A job shop is an organization equipped with several work stations, enabling the performance of operations on objects, (Grefenstette, 2014). Derived from this definition, the problem connected to the job shop is obtained.

”The job shop scheduling problem (JSSP) consists of a set of jobs on a set of machines with the objective of minimizing the makespan.”, (Yin et al., 2011). The job shop scheduling problem is one of the best known combinatorial optimization problems, (Graham, 1966). Job shop scheduling problems have been extensively researched, (Yin et al., 2011), (Zhang et al., 2017), driven by industries' need for more effective scheduling in order to stay competitive in the market. Establishing assumptions to the problem which is to be solved is essential in order to describe the basis for the modeling, in addition to simplifying modeling aspects which otherwise would hinder the models to run. Assumptions of the problem to be addressed are stated below, (Boushaala et al., 2012), (Bierwirth and Mattfeld, 1999):

- Each job has a predetermined unique sequence of operations to be completed in a specific order on a selection of machines
- All machines are unique
- Processing times for all jobs are constant and known
- An operation of a job can be performed by only one machine
- Each machine can perform only one operation at a time
- Operations cannot be interrupted
- An operation of a job cannot be initiated before the job's preceding operations are completed
- Transportation time between machines and changeover times are non-existent

A job shop consists of machines, [M_1, ..., M_m] which perform different types of service to n jobs, [J_1, ..., J_n]. Each job consists of at most m tasks, of which each task has a processing time denoted by [P_{j1},..., P_{jm}]. The task T_{j1} has to be completed by its designated machine M_i before task T_{j2} can commence; thus, all tasks have to be performed in a specified order.

Figure 3.3 depicts an example of a job shop schedule, consisting of three jobs, three machines, and three operations to be done for each job, in precedence.

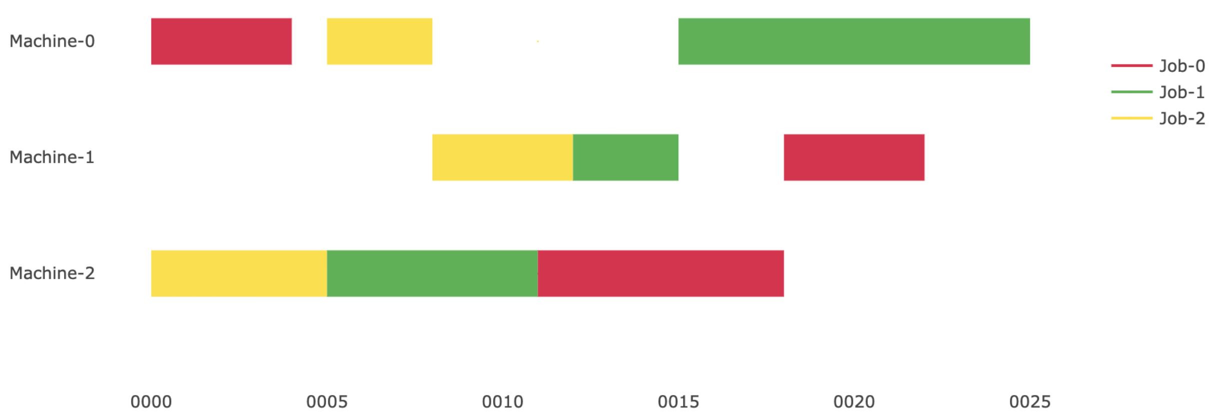

Figure 3.3: An example of a job shop schedule

3.2.2 Importance of job shop scheduling

The complexity of scheduling increases exponentially with the number of products and number of machines, rendering the act of determining a shortest-length schedule in an m-machine job-shop an NP-complete problem in instances where m $\geq$ 3, (Garey et al., 1976). The term NP-complete, explained in a simplified manner, defines problems of which the solution can be verified in polynomial time, but is as hard as the hardest NP-hard problem. This entails that although verifying a solution is feasible, finding the optimal solution may take forever using brute force methods, testing all possible solutions, with the computational power available today.

Scheduling is among the most essential tasks in the planning and operation of manufacturing companies, (Dahal et al., 2007). As obtaining optimal scheduling solutions entail computational challenges, manufacturers' production plans tend to bear the mark of sub-optimal schedules. Supply chains tend to act as though they perform better in their risk management capabilities, rendering them vulnerable to consequences this entails, (Rajesh, 2018). Effective job shop scheduling grants several benefits, capabilities which the production planner otherwise would be without.

A transparent job schedule grants visible planning and production processes, enabling the act of identifying bottlenecks. Effective scheduling enables the production planner to perform what-if analysis and simulations, avoiding to settle with just one production plan based on predictions. A manufacturing system is dynamic, prone to unexpected events, and therefore dependant on dynamic scheduling in order to reschedule an existing production schedule when the state of the manufacturing system renders it infeasible, (Vieira et al., 2003).

In the event of a postponement of an order, effectively delaying the delivery date of the order, either triggered by the customer or due to supplier problems postponing the final assembly, all future operations on the relevant machines may ideally be expedited. This would decrease unnecessary late deliveries for remaining orders, and thus increase the supply chain's measurement of delivery in full, on time (DIFOT). The functionality of rescheduling exists in some manufacturing companies but in far from all. Production planning is a time-taking process, argued to be prone to errors, and tend to be static, leading to delays as real-time changes are not communicated among entities of interest, (de Man and Strandhagen, 2018).

In literature, three common types of schedule repairs exist. Firstly, regeneration aims to perform a total reschedule of all jobs, not only jobs affected by disruptions. As all operations need rescheduling, this method is regarded as computationally expensive. Secondly, partial rescheduling only aims to reschedule jobs which are affected by disruptions. Thirdly, the right-shift-method aims to postpone all remaining operations by the value of the downtime caused by disruptions, (Vieira et al., 2003).

In order to cope with tasks such as efficient job shop scheduling, complex planning, and frequent rescheduling, it requires a sophisticated planning and scheduling system, using complex mathematical algorithms and logic to perform optimization, while considering multiple constraints. This type of system is commonly referred to as advanced planning and scheduling (APS) systems.

3.3 Advanced Planning and Scheduling systems

There are several definitions of the term advanced planning and scheduling (APS) systems, but the following definition constructed by (Naden, 2000) has been selected to represent the term, as it captures challenges proposed in this study's problem formulation:

"APS is a set of technologies, business processes, and performance metrics that enable manufacturing companies to compete more effectively in the global market place. The technologies involved are computer software and hardware that enable the organization to change the way they plan, schedule, forecast, distribute, and communicate with customer and suppliers".

In the 1990s, when technological advancements enabled the spawn of new planning concepts, made possible by increased computing power, APS became introduced, adding capabilities such as constraint-based planning, scenario simulation, multi-site production planning, finding feasible, near-optimal plans while potential bottlenecks are considered explicitly, and the adaption of advanced logic for capacity and supply chain planning, (Turbide, 1998), (Moon et al., 2004), (David et al., 2006), (Stadtler and Kilger, 2002). It was recognized that ERP systems were not satisfactory for the increasing need for product variety, on-time deliveries, and shorter lead times, (Henning, 2009), (Kristianto et al., 2011). Additionally, even advanced ERP systems fail to evaluate all effects on customer orders caused by disruptions and unpredicted events at the plant floor level. Several constraints, such as concerning materials, changeovers, and multi-tasks, are not taken into account, (Lupeikiene et al., 2014). Optimizing bottleneck processes are thus deemed to be an unfeasible task, (Nishioka, 2005).

Ever since the term APS was introduced, it has been referred to as different systems regarding which tasks they deal with. As the term still is regarded as new, only recently gained the attention of supply chains, the definition varies, (Hvolby and Steger-Jensen, 2010). Comparing APS systems to its predecessors may assist in shedding light upon the main contributions they provide.

3.3.1 Material Requirements Planning

Material Requirements Planning (MRP) systems are used to manage manufacturing processes, such as production planning, scheduling, and inventory control. MRP once represented a giant leap forward in material planning processes, utilizing a master production schedule, ran by a computer to calculate material requirements, in addition to a bill of material describing all materials required to produce a given product, (Umble et al., 2003). The production plans were produced in a two-step procedure. Firstly, material requirements were calculated, using the master production schedule, granting data regarding the production, in addition to the forecast, assuming infinite capacity. Furthermore, in the second step, the capacity was calculated. Materials and capacity were consequently planned separately, often resulting in unfeasible plans. In order to obtain a feasible plan, data required altering, accomplished by performing the planning procedure yet again, (Lupeikiene et al., 2014). Although the planning resulted in feasible plans, they were often far from optimized regarding operation sequencing, (Hoppe, 2007).

3.3.2 Manufacturing Resources Planning

Manufacturing Resource Planning (MRP II) is an extension of the original material requirements planning (MRP). MRP II systems evolved to not only containing manufacturing and materials management systems but to contain financial accounting - and management systems. MRP II granted supply chains a more integrated business system, (Umble et al., 2003). Material requirements planning was still an essential part of the approach, but now followed by sequentially executed steps such as capacity requirements planning, then scheduling (Hoppe, 2007). Due to the sequential execution of planning steps, MRP II is characterized as having a long planning duration, resulting in long planning cycles, causing production plans to become outdated. Planning and scheduling are still based on the availability of an infinite amount of resources, and the capacity utilization is weak, granting no ease in improving bottlenecks. Additional traits of MRP II are compared to corresponding traits of APS systems in Table 3.2.

Table 3.2: Comparison of MRP II and APS, adopted from (Van Eck, 2003)

MRP II system	**APS system**
Lead times are fixed	Lead times may be dynamically adjusted after communicating with customers
Usually batch sized with long durations	Dynamic establishment of plans and schedule
No support for decision making	Support decision making by enabling simulations and what-if analysis
Material allocation is handled on a first come first served basis	Material allocation is handled in regard to availability and specified criterion

3.3.3 ERP

Enterprise resource planning (ERP) systems provide a unified enterprise view of the business which encompasses all departments, and a database containing entered, processed, recorded and monitored business transactions, (Umble et al., 2003). The core of early ERP systems was the MRP system, described in Section 3.3.1. ERP systems are still to a large degree utilized today, either as a supply chain's primary enterprise system or by serving a function to APS systems. APS systems are often classified as complimenting existing ERP systems, enabling planning and scheduling functionality, or systems developed in-house to provide decision support. Some APS systems encompass planning processes at multiple planning levels, while other systems are tailored towards providing planning capabilities for one particular planning process, (Ivert, 2012). Extraction of data from an ERP system may serve as input to an APS system. Furthermore, the APS system can perform calculations of the data, and provide results back to the ERP system, (Stadtler and Kilger, 2002). However, a prerequisite for both systems to serve one another with data is that the systems are properly integrated. Changes made in one system must spark changes in related systems, as an absence of this results in unrealistic input data and constraints, and thus resulting in unfeasible production plans. Additional functionality provided by APS systems compared to ERP systems is described in Table 3.3.

Table 3.3: Comparison of ERP and APS, adopted from (Lütke Entrup, 2005) and (Infor, 2010)

Areas	**ERP system**	**APS system**
Planning philosophy	Assumes unlimited amount of key resources Push manufacturing Sequential and top-down Limited to planning only production areas where the ERP system is installed	Planning provides feasible plans in regard to available resources Pull manufacturing Integrated and simultaneous Can connect to multiple systems inside and outside the organization for global planning
Ability to optimize cost, price, profit	Not available	Available
Incremental planning	Not available	Available
Rescheduling capabilities	Low and slow	High and fast

3.3.4 Structure of APS

Several APS systems have been developed by independent software companies, and although definitions of APS are diverse, there exists a typical structure among them. APS systems commonly consist of numerous software modules, tackling different parts of the planning tasks, (Meyr et al., 2008). The different software modules are depicted in Table 3.4.

Table 3.4: Software modules, building bricks of APS, adopted from (Meyr et al., 2008). The highlighted module is most relevant for the scope of this thesis.

Time frame	Module	Responsibility
Long term	Strategic Network Design	Responsible for designing the supply chain, determining tasks such as the physical distribution structure to the material flows between suppliers and customers.
Mid term	Master planning	Responsible for coordinating production, procurement and distribution on the mid-term planning level.
Mid term	Demand planning	Responsible for conducting the strategic sales planning and mid-term sales planning. Parameters such as capacity, personnel and distribution is considered simultaneously. Master production scheduling is supported.
Short term	Purchasing & Material Requirements Planning	Responsible for handling purchasing planning for materials and components which an ERP system is unable to cope with, such as evaluating alternative suppliers, potential quantity discounts, as well as lower and upper bounds on supply quantities. However, planning tasks such as the BOM explosion and ordering of materials are the ERP system's responsibility, if an ERP system is present.
Short term	Production Planning & Scheduling	Responsible for determining lot sizes and the processes in production , regarding machines and shop floor control. In order to function efficiently, this module has to be designed to cope with bottlenecks and potential multi-stage operations. Due to the need to tailor this module to specific requirements, some vendors split this into two modules, one handling production planning and one handling scheduling.
Short term	Transport Planning & Distribution Planning	Responsible for planning the short-term transportation. A sub-module, ”distribution planning”, may be designed in order to handle material flows on a short-term level, more detailed than the Master Planning module conducts.
Short term	Demand Fulfillment & ATP	Responsible for planning the short-term sales, such as granting answers to questions regarding if a product is available or not.

The modules described in Table 3.4 may be bundled together in a complete solution or may be individually implemented one by one, according to the supply chain's needs. A supply chain's existing systems may cover some modules, resulting in a full implementation of all software modules unnecessary costly. However, bundling the APS modules with modules already existing in ERP systems is a common practice, resulting in the final product being an extensive supply chain suite, (Meyr et al., 2008).

Most relevant for the scope of this thesis is the module called Production Planning &

Scheduling, highlighted in Table 3.4, consisting of tasks belonging to the short-term planning tasks.

3.3.5 Modelling and evaluation of conflicting objectives

APS systems utilize complex mathematical algorithms and logic to schedule production and create production plans with certain constraints in order to optimize their objectives. In the event of several objectives present, a trade-off analysis is performed, and the fulfillment of one objective may be sacrificed for a greater profit in another objective. High customer satisfaction levels have argued to be essential in order to ensure customer loyalty and retention (Singh, 2006), but in order to prioritize customer satisfaction at all cost, inventory holding costs would become extensive.

Operating with constraints and capabilities in such a complex model enables the system to generate far more realistic and reliable production plans than creating them manually. An approach to modeling and evaluating conflicting objectives is required, and two options are argued to exist: constraint-based planning and optimization, (Steger-Jensen et al., 2014).

3.3.6 Constraint-based planning

Constraint-based planning operates with soft and hard constraints. The soft constraint may be overruled if necessary, while hard constraints may not be overruled. The distinguishing feature is that objectives may be described as to minimize deviations from specified goals. Since no criteria or plan optimization objectives are established, a feasible plan will be produced, but not necessarily an optimal one. As a result, APS systems utilize a hidden plan objective function influencing the planning by conducting trade-offs among the soft constraints, (Steger-Jensen et al., 2014).

3.3.7 Optimized plans

Optimized plans, however, are generated based on plan objectives, penalty factors, and constraints. Decision variables and penalty factors replace the constraint-based rules. The new metric for optimizing the entire plan is now the relation between profit and cost. Soft constraints may be overruled during the optimization if the total cost is calculated to be reduced, (Steger-Jensen et al., 2014). The objective function tends to consist of three parameters: on-time delivery, plan profit, and inventory turns, of which each can be weighted to the desired degree.

3.4 Artificial intelligence

The complexity of scheduling has exploded as the number of inputs and constraints to optimization problems have increased significantly, (Pishvaee et al., 2011). Increased computational power has granted the ability to solve previously impossible problems, and one major element assisting in solving these problems is the introduction of artificial intelligence (AI). Among the breakthroughs are advancements in medicine (Patel et al., 2009), innovations in cybersecurity (Dilek et al., 2015), and outperformance of world-class chess players in their expert field, (Bringsjord, 1998). Also, AI has been thoroughly researched in scheduling problems and has proven to be effective in solving well-known problem instances. A typical scheduling problem is the traveling salesman problem, consisting of a list of cities, distances between each pair of cities, and aiming to locate the shortest route which visits each city before returning to the origin city. The use of artificial intelligence has proven to be highly effective in obtaining solutions for such problems. An introduction to AI and the genetic algorithm used to develop the optimization models follows.

3.4.1 Evolutionary algorithms

”Evolutionary algorithms are based on models of organic evolution, i.e., nature is the source of inspiration”, (Back, 1996), and is in the field of artificial intelligence. Evolutionary algorithms model the collective learning process within a population of individuals, each of which representing a search point in the space of potential solutions to a given problem. The starting population is randomly initialized and evolves in an iterative process to better solutions by operators such as mutations, selections, and recombination of populations. The recombination mechanism mixes parental information and passes it to the next generation of populations, and mutation results in populations with innovative traits, similar to evolution in nature. General principles of evolution, which are encompassed in evolutionary algorithms, are depicted in Table 3.5.

Table 3.5: Evolutionary traits

Term	**Description**
Population	Set of many individuals
Diversity	Individuals have different characteristics
Inheritance	Characteristics are transmitted over generations
Selection	Individuals best suited to their environment produce offspring
Variation	Offspring are recombined and mutated elements of their parents
Survival of the fittest	Population gradually improve over generations

The traits described in Table 3.5 will be further described in a specific algorithm, belonging to the field of genetic algorithms. Evolutionary algorithms encompass genetic algorithms and more, and the field of genetic algorithms will be the topic of interest in the following sections.

3.4.2 Genetic algorithms

"Genetic algorithms belong to the class of artificial intelligence techniques, and they are based on Darwin's theory about 'survival of the fittest and natural selection", (Andresen et al., 2008)

John Holland introduced the field of genetic algorithms in 1960, and the field remained largely theoretical until the 1980s when the first international conference on genetic algorithms in engineering systems was held. Genetic algorithms are suitable for optimization problems in large, complex search spaces. They have proven to discover good solutions for high-dimensional problems, such as combinatorial optimization. Advantages with genetic algorithms are stated below, (Haupt and Ellen Haupt, 2004):

- Optimizes with continuous or discrete variables
- Handles a large number of variables
- Avoids local minimum by being able to optimize with complex cost surfaces
- Outputs a set of feasible solutions
- Not dependant on derivative information

The above-mentioned properties of genetic algorithms aim to tackle the challenges that complex optimization problems entail. The ability to handle continuous and discrete variables renders the algorithm generic, and thus not limited to only one type of input variables. Furthermore, as the number of solutions gets out of hand, and the search spaces are extensive, the property of not settling with a local minimum is highly beneficial. The advantage of rapidly reaching good solutions without searching through the whole search space of possible solutions is argued to be the most valuable characteristic, enabled by the iterative process of evolving the population. Areas in the search space which prove to be attractive are both beneficial and possibly inconvenient, as good solutions can be obtained at an early stage, but may also result in premature solutions, namely in local minimums. To avoid local minimum or maximum, depending on what the objective is, sub-populations may be utilized in order to diversify the population, and thus searching in different areas of the search space. Genetic algorithms are characterized by utilizing a set of possible solutions, called a population, to perform a parallel search of the search space. By performing genetic operators such as crossover, mutation, and selection on the current population, new generations, often referred to as children, are obtained.

The genetic algorithm of interest in this study is the artificial bee colony algorithm, simulating the foraging behavior of honey bees, first introduced in 2005, (Karaboga, 2005). It has since been the topic of multiple studies, depicting great strengths in solving optimization problems, (Karaboga and Basturk, 2008), (Gao and Liu, 2012), (Yin et al., 2011). The following section describes the algorithm.

3.4.3 Artificial bee colony

Swarm intelligence is a research field which has gained lots of interest from scientists for the past decades. Bonabeau, (Bonabeau et al., 1999), defined swarm intelligence as ”any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies”.

The collective intelligence of bee swarms consists of three requisites: food sources, employed foragers and unemployed foragers. Each *food source* has a fitness value, describing factors such as its proximity to the nest, the concentration of its energy, and the ability to extract this energy. All of these factors produce a single value; the fitness value, (Seeley, 2009). *Employed foragers* are bees which are exploiting a certain food source, and calculates its fitness value based on the factors described above. After that, the bees return to the hive and share this information among other bees with a certain probability. *Unemployed foragers* make up for the rest of the bees, and they consist of *scouts* and *onlookers*. The scouts search the environment surrounding the nest for new food sources, while onlookers await the employed bees. The onlookers, however, establish a new food source based on information obtained by the employed bees, when they decide to share it.

There has been an attempt in replicating the behavior of employed foragers, unemployed foragers, and the food sources by data scientists. Food sources represent a feasible solution to an optimization problem, and solutions which seem promising are modified by operators described in the previous section, namely crossover, mutation, and selection in an attempt to mimic the behavior of bees. Additional modifications are performed in order to enhance the individuals. Among them is the fast local search, in addition to a pairwise based local search. The latter method aims to enhance the global optimal solution and assist in escaping local minimums. Figure 3.4 depicts the algorithm’s logic and steps from the initialization until the result is produced. A further explanation of the various operators and search methods follows.

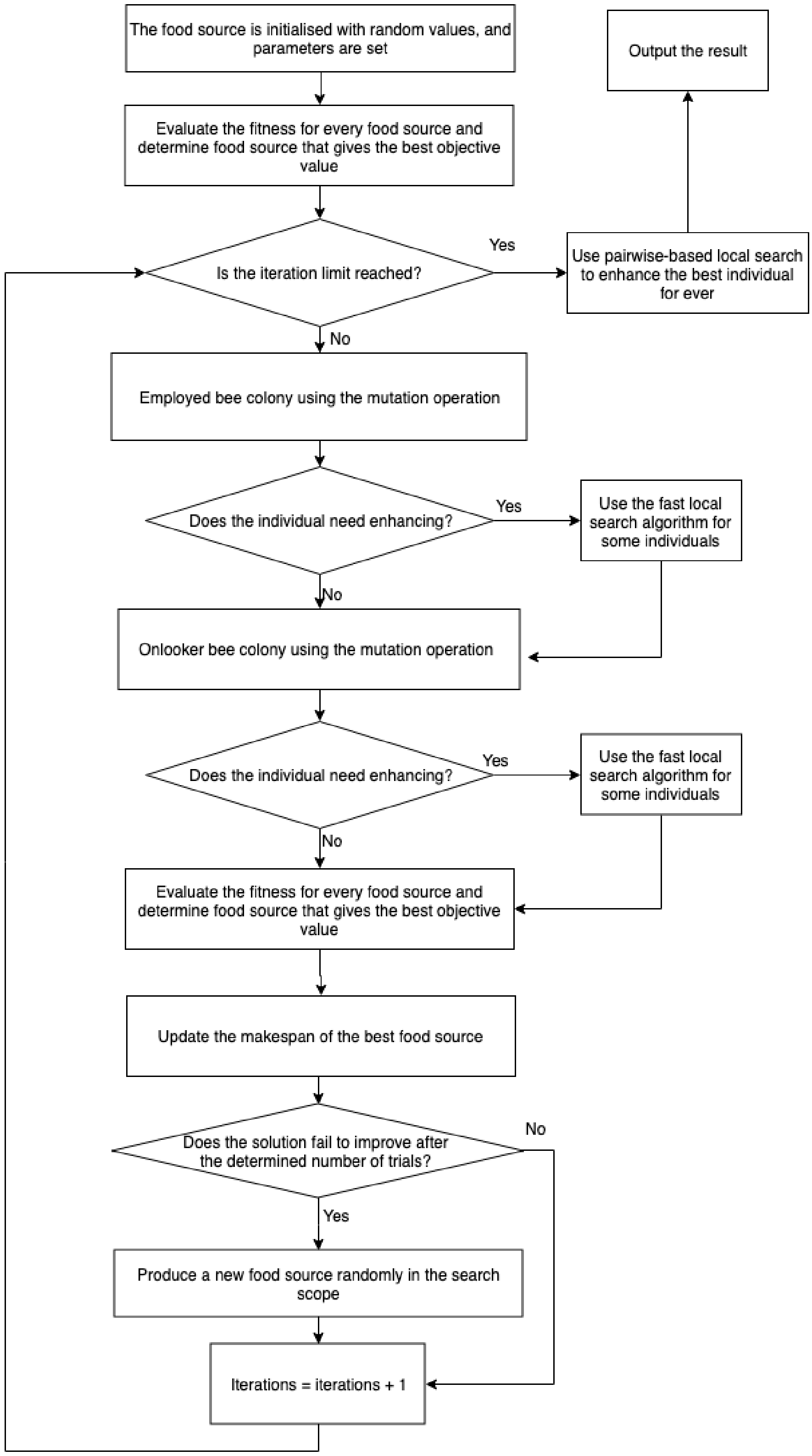

Figure 3.4: The framework of the proposed algorithm for job shop scheduling, adapted from (Yin et al., 2011)

3.4.4 Fast local search

The purpose of the local search is to find a better solution from the neighborhood of a solution. In order to improve the local search ability and obtain a more fitting solution, a new fast local search is proposed in order to decrease the makespan or tardiness of the food source. Three operators on job permutations are applied in order to enhance the individuals, namely, the swap operator, insert operator, and invert operator. The operators are initiated from an initial solution and aim to traverse from one solution to its' neighborhood, exploring new solutions. If the neighbor's objective function value is more desirable, the new solution is accepted as the current best. Whenever an improvement of the objective function value occurs, the orders of the jobs are permanently interchanged. This sequence of operations continues until criteria have been met, or until the specified number of iterations have been completed.

Swap

Randomly choose two different positions from a job permutation and swap them, depicted in Figure 3.5.

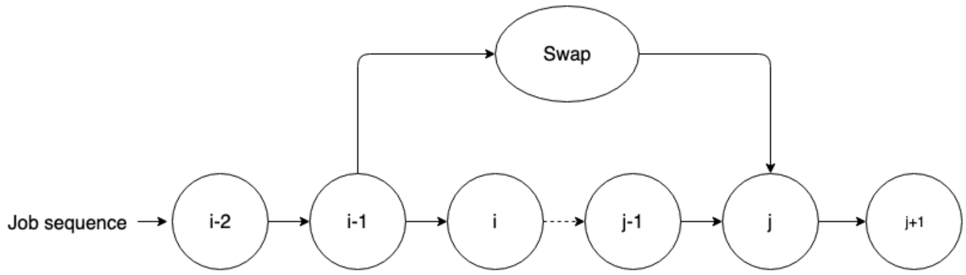

Figure 3.5: Swap operator

Insert

Randomly choose two different positions from a job permutation, and insert the former value behind the latter value, depicted in Figure 3.6.

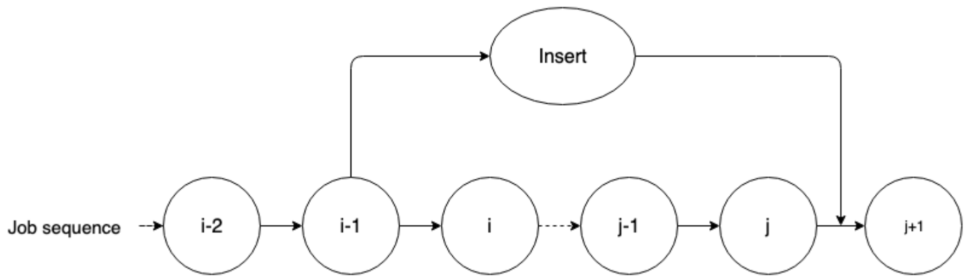

Figure 3.6: Insert operator

Invert

Invert the subsequent values between to random positions of a job permutation, depicted in Figure 3.7.

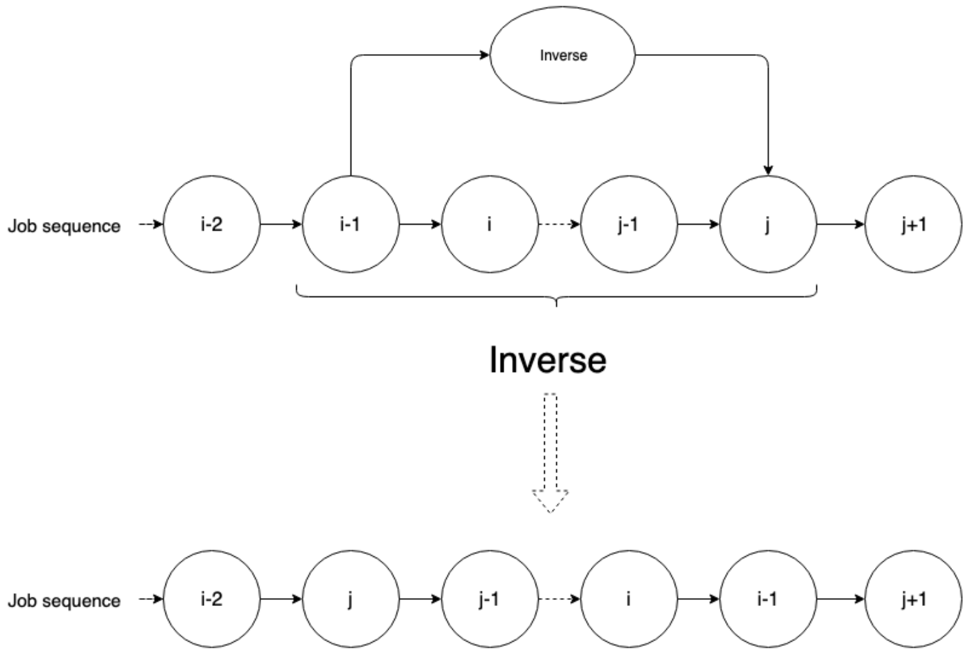

Figure 3.7: Invert operator

3.4.5 Pairwise-based local search

The purpose of the pairwise based local search is to aim for a better global optimal solution, developed by Aldowaisan and Allahverdi, (Aldowaisan and Allahverdi, 2003). The act of such a search results in selecting the best solution so far, then iteratively swapping job operations of two adjacent jobs of the individual and thereafter comparing the objective function value from the previous individual to the new one. If the new value is more fitted, the new individual is accepted. All job operations become subject to the swapping operation with adjacent orders in the hope of obtaining a better solution. This is proven to be an effective operation to escape local optimums.

3.4.6 Parameter tuning

The quality of the output from evolutionary algorithms greatly depends on the parameters' values. Due to computational limitations, values have to be selected carefully, (Karafotias et al., 2015). Parameter values are determined and set before running the evolutionary algorithm. Typical parameters for evolutionary algorithms are the population size, the selection mechanism, and crossover - and mutation rate. The algorithm used in this paper has the following parameters:

- population size
- iterations, the number of repeated operations
- maximum number of generations a food source can be attempted to be improved before it is discarded to a random food source
- rate to conduct a pairwise search
- crossover percentage on the population

There is often no upper limit to the number of values each parameter may have, resulting in the act of selecting reasonable values to be essential, computationally-wise. By limiting the different values for each parameter to six, the number of algorithm setup alternatives still become immense.

$$6^5 = 7776 \; unique\ parameter\ value\ combinations \tag{3.1}$$

$$7776 * 100 \; iterations = 7,776,000 \; runs \tag{3.2}$$

$$7,776,000 \; runs * 30 \; seconds = 2700 \; days \tag{3.3}$$

In order to establish the best results based on specific input data, testing all combinations of parameter values, would according to (3.3) take 2700 days, which is practically infeasible, and undoubtedly unpractical. Therefore, testing rough selections of the possible values is essential.

As genetic algorithms always maintain or improve their solutions, running the algorithm for an extended amount of time will never produce worse results. However, as argued, there is a limit to how long an algorithm may run before the modeling becomes impractical concerning time. Generally, the larger the value of the population size and iterations, the better the solution becomes; however, the required computational time to achieve solutions increases. A recognizable approach to this computational problem is to run the algorithm until the output converges, or until the amount of time allotted to it is exhausted, whichever first occurs.

Selecting values for the genetic algorithm parameters as described in the list of parameters above is a challenging task due to the algorithm relying on random number generation in several parts of the execution. By running the algorithm multiple times with a new random number generation for each run, the algorithm will produce different results each time. In order to best tweak the parameters in order to obtain valuable results, a fixed random seed is set. This ensures that the random generation is the same for each run, enabling the possibility to compare different runs to one another.

3.5 Summary of the theoretical background

Several fields of interest have been discussed in Chapter 3 in order to shed light on topics which are highly relevant for the development of the models, and the fulfillment of the research objectives stated in Section 1.3. Key takeaways from this chapter follow. The production environment in the engineer-to-order (ETO) production approach has been elaborated, identified as high-variety, low-volume (HVLV). Common traits of this production environment are complex product routings, a high degree of customization, low volume, and deadlines being essential to meet due to customer satisfaction. Furthermore, production planning and scheduling in the same environment have been discussed, introducing job shop scheduling as the appropriate type of schedule for the development of the models, in addition to being relevant for the corresponding case company data. Moreover, the complexity of job shop scheduling arises a requirement for systems utilizing advanced logic to produce feasible and close-to-optimum schedules. This type of system is identified as advanced planning and scheduling (APS) systems and is introduced in order to compare the developed models to the system's core optimization model. Lastly, as the developed models are based on aspects of artificial intelligence, knowledge regarding the field, in addition to subsets of the topic such as evolutionary and genetic algorithms, assists in understanding how the models' results were obtained.

Chapter 4

Model development

4.1 Approach and assumptions

The model which had been developed was required to tackle several sets of assumptions and objective functions, as characteristics of the company data differ from data which the genetic algorithm is based on, identified in (Yin et al., 2011). Unique characteristics of the three developed models are described in the following sections, and common assumptions for all models are as follows.

- All machines are unique
- Processing times for all the jobs' operations are known
- An operation of a job can be performed by only one specific machine
- Jobs do not visit a specific machine more than once
- Each machine can perform only one operation at a time
- Operations cannot be interrupted
- Transportation time between machines is non-existent
- No machine breakdowns occur
- Job-specific setup times or machines' changeover times are not considered
- The model performs scheduling based on batches of orders

Due to the existence of dynamic manufacturing environments, as stated in Chapter 3, incremental planning, a two-step planning procedure, has been enabled in the developed models. Continuous estimation of delivery dates of proposed customer orders and evaluation of whether customer requests are feasible or not can be carried out at any time by generating what-if scenarios. If there exist time gaps in the current schedule such that only minor adjustments are necessary in order to handle new orders, the acceptance of new orders can proceed. However, a comfortable freezing period is advised, rendering planned production static for a given amount of time in order to avoid abruptly disrupting continuous production. This function can be further exploited by rescheduling all remaining activities, avoiding the need to search for time gaps and fit jobs in-between others.

4.2 Unique characteristics of three variations of the algorithm

Three variations of the algorithm have been developed, each to optimize based on unique objectives. The first algorithm optimizes based on the makespan, similar to the paper which proposes the genetic algorithm, (Yin et al., 2011). The second algorithm optimizes based on unique characteristics of the case company, which differ from the assumptions of the first developed algorithm. Furthermore, the third and last algorithm optimizes based on a combination of the identified characteristics in the research paper by (Yin et al., 2011) and in the case company's environment. The unique characteristics for each algorithm are stated in the following subsections.

4.2.1 Unique characteristics of the first algorithm

The algorithm of which the first developed model is based on optimizes the production schedule on the aforementioned assumptions, in addition to the following unique assumptions:

The objective function is defined only to minimize the makespan. The first algorithm assumes that each job has a predetermined unique sequence of operations to be completed in a specific order on a selection of machines. A given job's operation cannot be initiated before the job's preceding operation is completed.

4.2.2 Unique characteristics of the developed algorithm for the case company

The case study is presented in Chapter 5, but the key characteristics of the company's production environment are presented here, in the context of the model development.
When selecting an appropriate optimization algorithm, it was assumed that the company's data would resemble data used with the original model. The most significant difference is the lack of precedence of jobs. While the original model schedules jobs' operations in a specific order, the case company manufactures several parts for the same product in parallel, in great contrast to the original model. However, the case company's data share similarities regarding the machines utilized for the manufacturing process, and the model can therefore be utilized to produce results of interest. The developed model has been modified in several areas to handle data made available by the case company. The most beneficial optimization criteria for the case company is minimizing the number of orders delayed, and therefore, jobs are scheduled according to the delivery date.

Two different objective functions are established in order to produce two different schedules for the same input data. The first objective function aims to minimize the number of delayed orders, in addition to minimize the inventory holding. Therefore, the jobs are scheduled such that the time between a job's completion time and the job's deadline is minimized, without exceeding the deadline. This property can be costly in the event of delays, as the jobs have no slack time. The completion time margin for each job can, therefore, be adjusted according to objectives, creating a buffer such that minor delays may occur without resulting in the job exceeding the deadline.

The second objective function, in contrast to the model described in Section 4.2.1, aims to minimize the number of delayed orders in addition to minimizing the makespan. However, the most significant part of this objective function is to minimize the number of delayed orders. Therefore, when all jobs are within their deadlines, further optimization on the total makespan ensues. As jobs are complete in the shortest time possible, the slack time for each job will tend to be considerably more comfortable than for the first objective function mentioned previously. Advantages and disadvantages for both objective functions are discussed in Section 7.2.

4.2.3 Unique characteristics for the further developed algorithm

As the model described in Section 4.2.1 excels with scheduling in accordance with precedence of operations, other production environments than the case company's have been evaluated. The model is further developed to solve multiple jobs, multiple operations on multiple machines using an objective function encompassing both deadline and makespan. The further developed model can accept input data similar to the model in Section 4.2.1, but can now also accept deadlines for each job. The developed model forces the resulting production plan to satisfy deadlines. The schedule's makespan is therefore negatively affected, as an optimization model minimizing only on makespan would produce a lower makespan value. However, this trade-off may be argued to be beneficial, as exceeding orders' deadline penalizes the supply chain in several areas, as argued in Section 1.2, and negatively affects the fundamental measure of delivered in full, on time (DIFOT). Input and schedules encompassing this logic are depicted in Section 6.3.

4.3 Input parameters to the optimization models

Several input parameters are required to run the optimization models. All elements shown in Table 4.1 exist in all three variations of the optimization model, except for index D, the due date, which is not existent in the model proposed in Section 4.2.1, and the index s, the successive operations, which is not existent in the model proposed in Section 4.2.2. A description of each input parameter follows, (Yin et al., 2011):

Table 4.1: Input parameters to the optimization models

Indexes	Description	Possible values
i	Iterations	[1, 10^5]
SN	Population size	[1, 10^3]
FS	Maximum number of food source generations	[1, 10^3]
m	Number of machines	[1, 10^2]
j	Number of jobs	[1, 10^2]
p	Processing time	[1, 10^9]
o	Operations	[1, 10^2]
s	Successive operations	[1, 10^2]
D	Due date	[1, 10^4]

- Iterations represent the number of times the algorithm performs operators to the input data, aiming to produce better results. Running the algorithm for numerous iterations will never weaken the result; thus, the higher the value for this parameter, the better the result will become. However, the computational time to obtain the result is proportional with the number of iterations, so the optimization should come to a halt when the objective value converges, or the computational time becomes extensive, as described in Section 3.4.6.

- Population size describes the number of food sources, representing a number of SN individual solutions which are subject to operators for each iteration. The larger the population size is, the greater number of solutions are evaluated for each iteration, and thus extending the required computational time.

- The maximum number of food source generations describe the number of times operators can mutate a food source before it is discarded to a random food source. Reverting a food source to a random value is beneficial such that local minimums are discarded, aiming to obtain an even better solution.

- The number of machines describe the number of machines which perform operations to jobs in a given data set.

- The number of jobs describes the number of jobs which are present in a given data set.

- The processing time describes the amount of processing time each operation in each job requires.

- The number of operations describes the number of operations each job requires.

- Successive operations depict the precedence of a job's operations, which cannot be interchanged.

- The due date describes the value of the deadline, a value which is considered in two of the developed optimization models. The two models aim to satisfy all the jobs' deadlines.

4.4 Summary of the model development

A collection of common traits among all the three cases have been stated, in addition to specific characteristics of the original algorithm, of the case company's data, and of the further developed model. The initial optimization model was developed based on the common traits and thereafter adjusted to cope with new data and new objective functions. The first model, as described in Section 4.2.1, handles multiple jobs' operations in precedence, and optimizes solely based on the makespan. The second model, as described in Section 4.2.2, handles multiple jobs, whose operations may be produced in parallel, and optimizes based on minimizing the number of delayed jobs, in addition to either the total makespan or completing the jobs as late as possible, without exceeding the deadline. The third model, as described in Section 4.2.3, handles multiple jobs' operations in precedence, in addition to deadlines, and optimizes based on minimizing the number of delayed jobs, in addition to the makespan.

Chapter 5

Case study

It is deemed beneficial to make use of a case company when developing a generic model in order to relate the models up to real data. The case company is made anonymous as some information is regarded as sensitive.

5.1 Background

The case company is a single-source supplier and takes full responsibility for the manufacturing of equipment to the maritime industry. The company's environment is defined as high-variety, low volume, depicting their complex product mix, and large deliveries. A single product may consist of more than fifty components to be manufactured and assembled at their factories over the course of several months. Due to the increasing market competition, the company has introduced an increasing rate of new products, resulting in an increased production operation complexity.

As a manufacturer of maritime equipment, the business is profoundly affected by changes in - and reliant on - the maritime industry. Equipment to the maritime industry is regarded as a conventional engineer-to-order industry, (Gosling and Naim, 2009), thus affecting the company's production strategy and placement of their customer order decoupling point (CODP). As the production at the case company is based on a combination of an ETO and MTO strategy, with a high degree of customization enabled, the possible product variants are very many, resulting in a rapidly changing production environment. Furthermore, the shop floor layout is a combination of a fixed-position layout and cell layout, contributing to a high material flow complexity.

5.2 The case company's planning systems

Production planning is currently performed in Mircosoft Excel spreadsheets, in assistance with their enterprise resource planning system and their product lifecycle management system. The company acknowledges that there is functionality available in the systems used today, which is not yet adopted.

Increasing the use of IT and integrating IT solutions are part of the case company's priorities in order to approach Industry 4.0. Decision support is greatly influenced by data analysis, which is enhanced by more integrated systems and thus an increase of seamless flow of information. Order data such as the products' processing time, the degree of work-in-progress, delivery dates for the products, and to what degree the orders are on track to meet the delivery date, are collected. This data is regarded as essential to perform monitoring of project processes.

5.3 Characteristics of the case company

The case company's production process consists of three parts. Firstly, the adjustment of raw material takes place. Secondly, the manufacturing on machines takes place. Thirdly an assembly takes place. Among the three parts of the production process, the actual manufacturing is the greatest cause of delays. Following the increase in customized products, the case company has a greater focus on shifting the CODP upstream. Being involved with customers during the customization period positively affects customer satisfaction, but also has some drawbacks.

Characteristics of the case company are stated below:

- The production strategy and CODP placement is a combination of ETO and MTO
- Due to the production strategy, a large number of product variants are offered
- Products are customized to a great extent
- The average throughput time varies from weeks to several months, depending on the product complexity
- The production strategy of ETO and MTO results in infrequent order repetition, resulting in complex and unique production routings.

5.4 Production planning until recent years

Until recent years, the case company developed a model aiming to accept different sizes of order units into the production plan based on characteristics such as production space allocation and work to be done. The capacity planning took place without considering assembly and wielding, as it was a great challenge to accurately establish constraints for these jobs. Establishing constraints for wielding is a challenge due to the flexible capacity of workers. An investment in additional wielding machines would render the constraints too flexible to be accurately set. Assembly has been argued to be an element which also is challenging to establish constraints for. Therefore, all the planning was revolved around the machines and their processing times. Large units were the first orders to be accepted to the plan, and when the production planners were unable to fit another large unit in the week's plan, smaller units were accepted to fill the capacity.

The Excel model was utilized until recent years until demand shifted to additional larger units and projects. The heuristics of small units could no longer be followed, as the company manufactured significantly less of those type of units.

5.5 Production planning today

Today the case company operates with rough-cut capacity planning, (RCCP). In the early planning phase, they duplicate an existing project to see the balance and move it in time in order to make capacity feasible. As the product mix is highly diverse, there are always changes to be done to the plan, such as additional wielding and adaptions. RCCP approaches are generally simple to implement, but there is a tendency that the estimated capacities may not be established accurately, such that non-feasible production plans are created. While the rough-cut capacity plan suggests that their capacity is sufficient, short-term capacity may be of shortage.

A typical rough-cut capacity planning variable is the defined capacity which is available for a given period, (Chua et al., 2011).

$$C_{available} = T_k U_k E_k$$

T_k refers to the total number of machine hours available at the work center for the defined planning period, thus the number of machines multiplied by the number of days in the planning period, multiplied by the number of operation hours a day. U_k refers to the machine utilization ratio for the work center, defining the percentage of operational hours to the total number of available hours. E_k refers to the machines' efficiency in the work center. This figure is obtained by dividing standard hours by hours work. Standard hours are defined as the amount of time required to set up a given machine and run one part through that given operation.

However, company data depicts that the machines do not run for the maximum number of hours on a weekly basis, since some machines are hardly part of any production plan, while other machines' usage approaches the full capacity.

The machines' efficiency, E_k, has been argued to be one source to the delays which the case company experiences.

5.5.1 Challenges with today's planning

The actual production tends to constantly be three weeks behind the proposed plan. The reasons for the delay are not certain, but some causes have been identified. The act of defining process times is subject to inaccuracy due to the product routings' complex characteristics. This results in tardiness, which calls for additional production planning, resulting in further delays. Additionally, the occurrence of missing parts for a planned assembly may result in added tardiness for the whole product. In the event of rush orders, spawned either by a priority order or due to quality problems of manufactured products, all products belonging to the machine routing of the rush order will be postponed, naturally resulting in delays.

Investing in additional machines which are part of the critical path in order to reduce the drag of each activity in the project can be argued to decrease the overall duration and increase the key performance indicators, but the production characteristics make this solution less feasible. Physical space to place new machinery is limited, and the investment is expensive. More importantly, the production from week to week and month to month are seldom fixed, resulting in different critical paths for each planning period due to the diverse product mix. Decreasing the estimated critical path is therefore challenging.

Chapter 6

Results and analysis

6.1 Results from the first developed model

The second research objective, as stated in Section 1.3, was to use a genetic algorithm based on artificial bee colonies, simulating the foraging behavior of honey bees. The algorithm of which the developed model is based on aimed to minimize the makespan of jobs, scheduled in accordance with assumptions stated in Section 3.2.1. The developed model indeed succeeds in scheduling jobs with operations in sequence, obtaining a lower and lower value of the makespan the longer the algorithm runs. However, as stated in Section 3.4.6, running the algorithm for an infinite amount of time is neither feasible nor practical, as advancements of the solution decrease exponentially when the solution approaches the optimal result. Therefore, the computation is halted when the value converges, or when the computational time becomes extensive, whichever comes first.

Test instances of job shop scheduling problems exist online, (Mattfeld, 2012), available for researchers to test the performance of their optimization models. The input data consists of a selection of jobs, with each job consisting of a number of operations to be run on the same number of machines in a specified sequence, with corresponding processing times. The test instances are denoted by a value describing the number of jobs to be scheduled, followed by the letter *x*, then a value describing the number of machines in sequence for each job. Therefore, *6x6* describes a test instance of six jobs, each job requiring six operations and six machines to handle the operations. The sequence of operations on machines are described in Table 6.1 and their processing times are described in Table 6.2.

Job 1's operations in Table 6.1 have the corresponding machines: 2, 0, 1, 3, 5, 4, describing the sequence of which machine each operation of Job 1 shall be executed on. This sequence is fixed and can not be interchanged.

Table 6.1: Jobs with corresponding machine types for each operation, in sequence

Job	**Machine sequence**					
1	2	0	1	3	5	4
2	1	2	4	5	0	3
3	2	3	5	0	1	4
4	1	0	2	3	4	5
5	2	1	4	5	0	3
6	1	3	5	0	4	2

Similarly, in Table 6.2, Job 1's operations have the corresponding processing times: 1, 3, 6, 7, 3, 6, where the first operation to Job 1 has a processing time of 1, followed by a processing time of 3 for the second operation, and so on.

Table 6.2: Jobs with corresponding processing times for each operation, in sequence

Job	**Processing times sequence**					
1	1	3	6	7	3	6
2	8	5	10	10	10	4
3	5	4	8	9	1	7
4	5	5	5	3	8	9
5	9	3	5	4	3	1
6	3	3	9	10	4	1

The data depicted in Table 6.1 and Table 6.2 become input data to the first optimization model. Figure 6.1 depicts the production schedule obtained, optimized regarding minimizing the makespan. The different colors of the Gantt chart represent a job, according to the list to the right of the schedule.

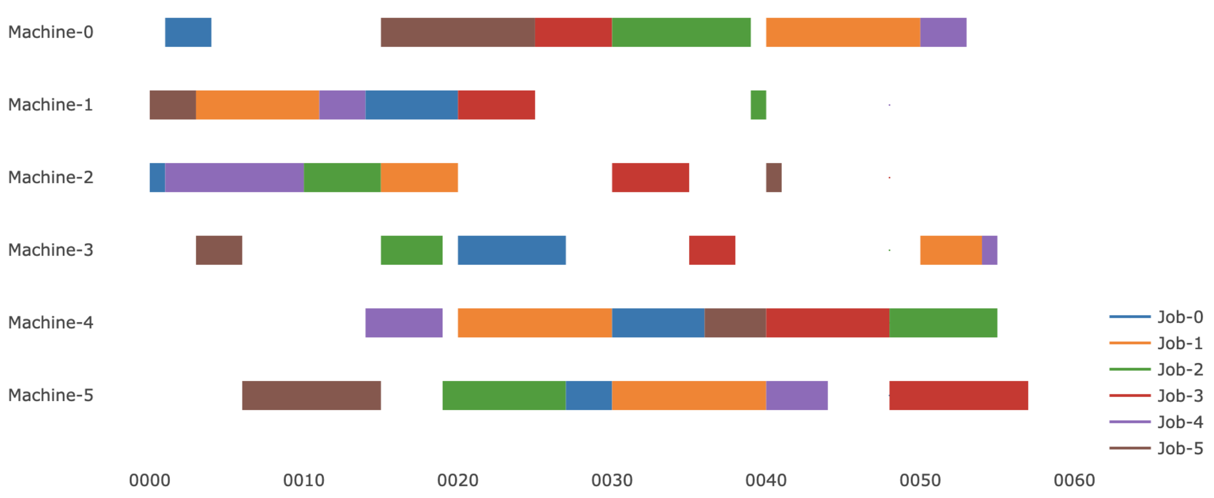

Figure 6.1: Job shop schedule, 6x6, with objective function to minimize makespan

Researchers have been testing the test instances on a selection of models for longer than a decade, and the best-known solutions to the test instances have been documented, available for comparison to other optimization models, among this one.

A *6x6* test instance is regarded as a small problem, among the test instances which are available for testing. Exponentially more computationally demanding are test instances operating with 20 jobs each consisting of 10 operations, manufactured on 10 machines, denoted as *20x10*. The search space, namely the feasible region defining the set of all possible solutions, in such problems becomes immense, and traversing through all possible solutions is not possible, as argued in Section 3.2.2. Utilizing an effective and efficient algorithm is necessary when handling such test instances, and genetic algorithms excel in obtaining good solutions in such cases.

Job 1's operations in Table 6.3 have the corresponding machines: 8, 7, 6, 9, 2, 1, 5, 4, 0, 3, describing the sequence of which machine each operation of Job 1 shall be executed on. This sequence is fixed and can not be interchanged.

Table 6.3: Jobs with corresponding machine types for each operation, in sequence

Job	Machine sequence									
1	8	7	6	9	2	1	5	4	0	3
2	4	5	3	9	0	8	6	7	2	1
3	5	4	2	6	1	7	0	3	9	8
4	1	5	0	3	2	7	8	6	9	4
5	2	5	6	9	1	3	8	0	7	4
6	1	4	0	2	9	8	5	3	7	6
7	5	9	0	4	6	3	2	1	8	7
8	5	9	8	7	4	6	3	0	1	2
9	1	8	0	2	9	3	5	6	4	7
10	4	3	6	5	2	8	1	9	7	0
11	4	7	9	2	3	8	5	6	1	0
12	8	5	1	7	2	3	6	9	4	0
13	2	4	3	1	8	6	7	0	9	5
14	0	8	3	7	5	2	4	6	1	9
15	9	0	4	8	6	2	5	3	7	1
16	3	2	5	0	7	4	8	1	6	9
17	1	7	8	3	4	5	6	0	2	9
18	1	7	2	0	8	6	3	9	5	4
19	2	3	4	9	0	6	7	8	1	5
20	1	0	5	3	9	7	8	2	6	4

Similarly, in Table 6.4, Job 1's operations have the corresponding processing times: 52, 26, 71, 16, 34, 21, 95, 21, 53, 55, where the first operation to Job 1 has a processing time of 52, followed by a processing time of 26 for the second operation, and so on.

Table 6.4: Jobs with corresponding processing times for each operations, in sequence

Job	Processing times in sequence									
1	52	26	71	16	34	21	95	21	53	55
2	55	98	39	79	12	77	77	66	31	42
3	37	92	64	54	19	43	83	34	79	62
4	87	77	93	69	87	38	24	41	83	60
5	98	25	75	77	49	17	79	44	43	96
6	7	61	95	35	10	35	28	76	95	9
7	59	43	46	28	52	16	59	91	50	27
8	9	43	14	71	20	54	41	87	45	39
9	28	66	78	37	42	26	33	89	33	8
10	96	27	78	84	94	69	74	81	45	69
11	24	32	25	17	87	81	76	18	31	20
12	90	28	72	86	23	99	76	97	45	58
13	17	98	48	46	27	67	62	42	48	27
14	80	50	19	98	28	50	94	63	12	80
15	72	75	61	79	37	50	14	55	18	41
16	96	14	57	47	65	75	79	71	60	22
17	31	47	58	32	44	58	34	33	69	51
18	44	40	17	62	66	15	29	38	8	97
19	58	50	63	87	57	21	57	32	39	20
20	85	84	56	61	15	70	30	90	67	20

The jobs, machines, and processing times described in Table 6.3 and Table 6.4 were input data, subject to the implemented optimization model. The algorithm was run with the following parameters shown in Table 6.5.

Table 6.5: Input parameters to the optimization model used to solve the *20x10* test instance

Indexes	Description	Values
i	Iterations	200
SN	Population size	90
FS	Maximum number of food source generations	45
m	Number of machines	10
j	Number of jobs	20
p	Processing time	[10, 99]
o	Operations	10
s	Successive operations	10

Figure 6.2 depicts the obtained schedule to the input data from Table 6.3, Table 6.4 and Table 6.5.

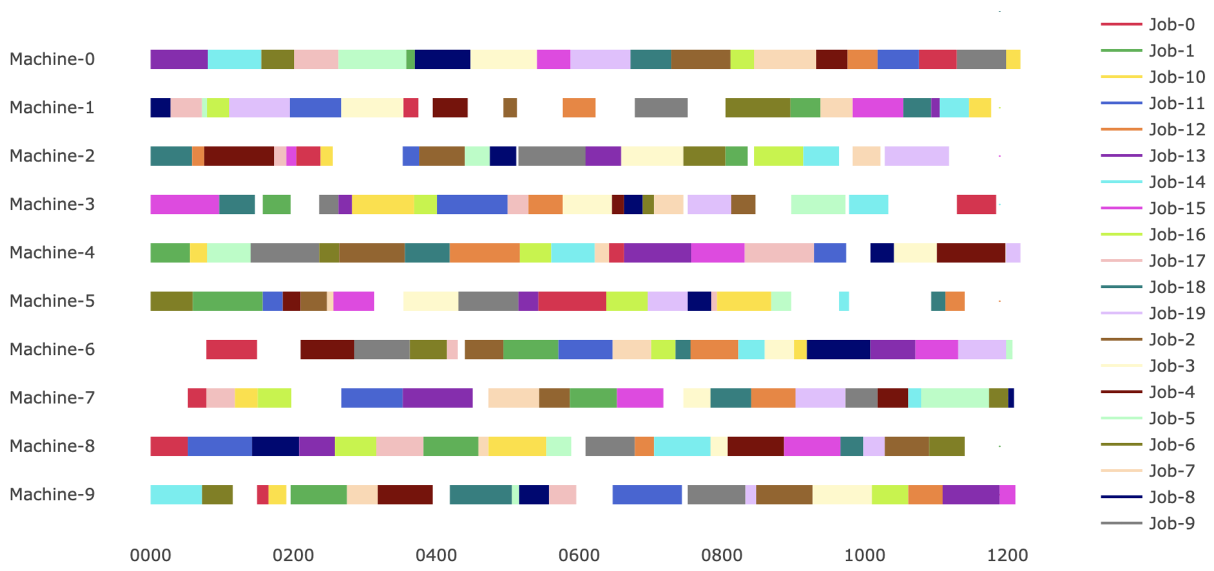

Figure 6.2: Job shop schedule, 20x10, with objective function to minimize makespan

As the Gantt chart consists of 20 jobs on 10 machines, following the flow of each job's 10 operations is quite the challenge. Key takeaways from the following Gantt charts will be mentioned in the subsequent paragraphs.

As described in Section 3.4.6, an obtained solution will never be discarded for a worse solution; thus, for each iteration, the solution will either remain at a specific value or be improved. The makespan for each of the 200 iterations is depicted by Figure 6.3. As can be seen from Figure 6.3, the first obtained solution can be regarded as having a makespan of approximately 1328, continually improving for each iteration. The significant improvement at 200 iterations was sparked by the pairwise-based local search presented in Section 3.4.5, a search method which is argued to be effective in escaping local minimums. In this test instance, the operation proved its efficiency.

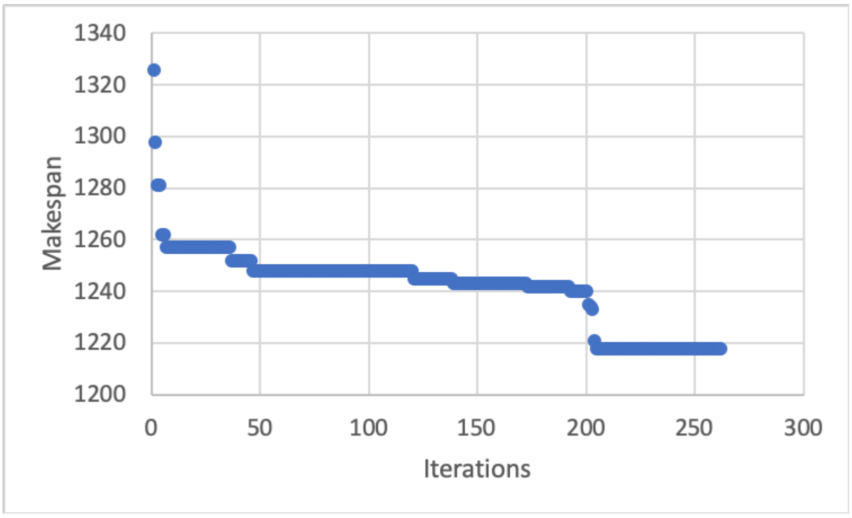

Figure 6.3: The obtained makespan after each iteration, converging to a value of 1218 after 205 iterations

Specific data of the two schedules obtained are described in Table 6.6.

Table 6.6: Results obtained using the first developed model

Problem size	Best known solution	Solution obtained	Computational time spent
6x6	55	57	0.7 seconds
20x10	1218	1218	1147 seconds

As argued in Section 3.4.6, a recognizable approach to computationally challenging problems is to run the algorithm until the output converges, or until the amount of time allotted to it is exhausted, whichever first occurs. In this case, with the *20x10* test instance, the computational time spent is 1147 seconds, and the makespan converges to 1218, the same value as the best-known solution to this specific problem, as stated in Table 6.6.

6.2 Results from the model tailored the case company

As described in Section 2.3, information regarding the case company was obtained through various sources. Among the collected data was a selection of products they manufacture in a given period of time. Four products were extracted from the data set and altered in order to run in the developed model. The case company's production environment differs from the definition of a jobbing process, which is described in Section 3.1. The optimization model considers step two of the three steps to the company's production process, as described in Section 5.3, namely the manufacturing process. At this step, the company's products consist of several components requiring only one operation on a single machine, thus making it possible to produce all components in parallel. This production sequence is in great contrast to the production sequence proposed in Section 6.1, due to the lack of precedence among operations. The processing times of the four products, in addition to deadline values, are shown in Table 6.7.

Table 6.7: Required machines, processing times, and deadlines of four company products

	Job			
	1	**2**	**3**	**4**
Machine 1	935	500	935	495
Machine 2	115	300	115	600
Machine 3	30	200	30	35
Machine 4	465	460	465	660
Machine 5	425	530	425	1350
Machine 6	57	262	57	420
Machine 7	965	532	965	750
Machine 8	45	743	45	2300
Machine 9	425	236	425	400
Machine 10	-	-	-	4533
Machine 11	170	754	170	215
Machine 12	400	123	400	1550
Machine 13	45	865	45	820
Machine 14	100	234	100	850
Machine 15	-	-	-	100
Machine 16	325	432	325	440
Machine 17	400	286	400	200
Deadline	1000	1700	2700	5600

Although the company's production sequence is in contrast to the production sequence proposed in Section 6.1, the company data can be subjected to the optimization model proposed in Section 6.1. However, an additional feature of this optimization model is the existence of deadline data for each job. Figure 6.4 depicts the obtained schedule by the developed optimization model, aiming to minimize the number of delayed orders, in addition to minimizing the makespan.

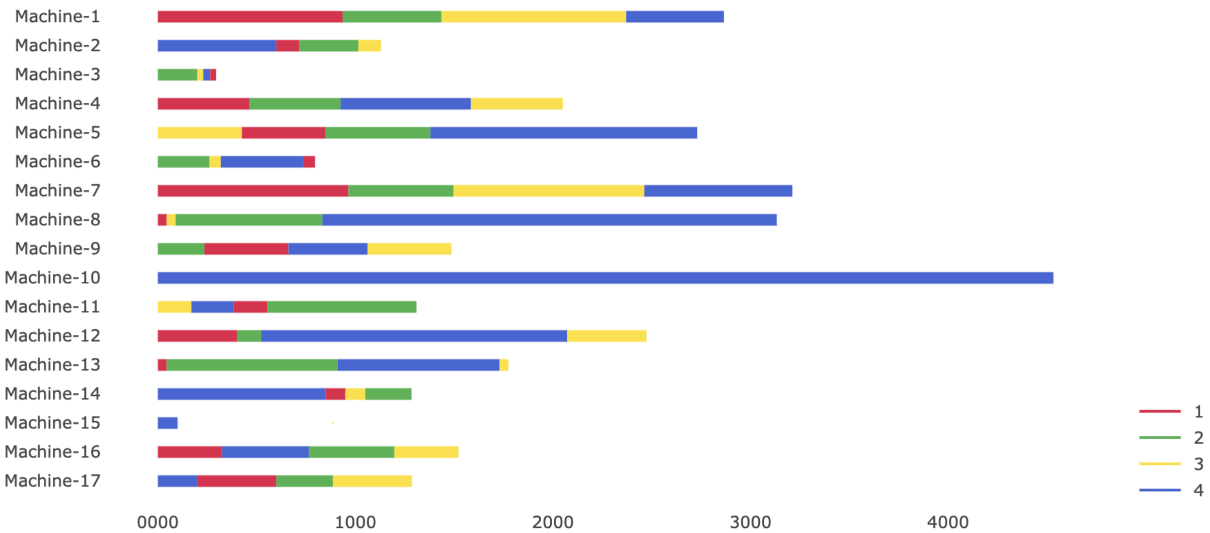

Figure 6.4: Results to input data from the case company, minimized in regard to makespan, constrained by deadlines

Figure 6.4 does indeed minimize the makespan, in addition to avoid exceeding deadlines. Production plans may have different objectives, as stated in Section 3.3.5. Makespan is argued to be the most common evaluation criteria, but additional criterion may be argued to be of greater importance at times. The objective of meeting deadlines has been argued to be of great significance in Section 3.2, and trade-offs have been discussed in Section 3.1, shedding light on inventory costs compared to meeting deadlines. In an attempt to decrease inventory costs, completion dates of products can be planned such that they are complete close to the deadline. Figure 6.5 depicts the company data scheduled by the developed optimization model, with each job's completion time delayed until the deadline.

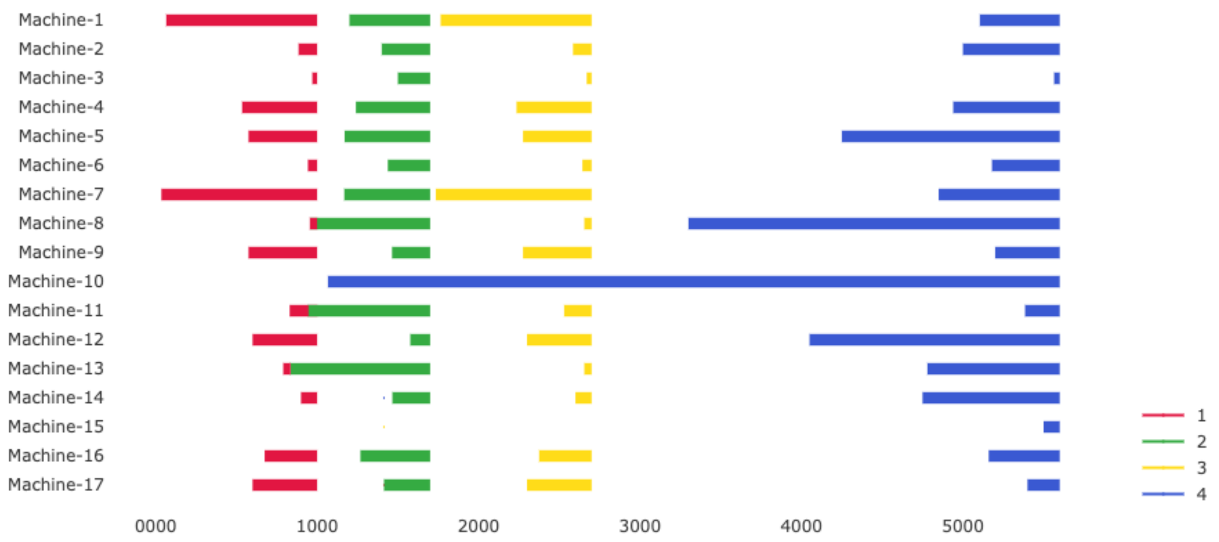

Figure 6.5: Jobs completed as late as possible, without exceeding the deadline

The jobs' slack, defined as the amount of buffer time each job can be delayed by without exceeding the deadline, decreases as the planned completion time approaches the deadline. Therefore, an evaluation as to what extent the completion time shall approach the deadline date is required to avoid exceeding deadlines due to disruptions. It can be seen in Figure 6.5 that some operations are completed before the deadline, in events where subsequent operations on the given machine require an extensive amount of time. A certain degree of storing components is therefore inevitable if deadlines are not to be exceeded.

6.3 Results from the model tailored industries with precedence among jobs' operations

The results described in Section 6.1 depict several jobs, each consisting of several operations, to be scheduled on several machines, in an attempt to minimize the makespan. The combination of multiple jobs, precedence among multiple operations and multiple machines to execute the operations is the primary source to the computational difficulties of job shop scheduling problems. This characteristic is not to be found in the production environment of the case company.

However, there are industries where these characteristics are present, industries operating with jobbing processes, as described in Section 3.1, where deadline is an essential criterion to optimize on. The model used to obtain results from Section 6.1 has therefore been further developed to handle deadlines. The test instances' data is altered by adding a value for each job, describing the deadline of which all of a particular job's operations should be completed. The specific values have been selected such that a schedule not considering deadlines at all, but solely based on the makespan, would exceed the deadline. The algorithm's logic is quite identical as to the one used to obtain results in Section 6.1, but it now strives to prioritize jobs with the earliest deadline by penalizing schedules if the jobs' completion times exceed the jobs' deadlines. The deadline is implemented as a soft constraint, thus still producing a plan if some jobs are unable to be completed within the deadline. However, rapid and frequent rescheduling is available to reschedule in an event such as a job exceeding its deadline, as described in Section 3.2.2.

6.3.1 Optimizing the schedule based on makespan and constrained by deadlines for the *6x6* test instance

The same test instances used for Section 6.1 are used for demonstrating the optimization model using the genetic algorithm with regards to the deadline. In addition to a specified sequence of machines and processing times as depicted in Table 6.8, there has been added a value for the deadline, depicted in Table 6.9 for the smallest test instance.

Table 6.8: Original input data for the *6x6* test instance

Job	Processing times sequence					
1	1	3	6	7	3	6
2	8	5	10	10	10	4
3	5	4	8	9	1	7
4	5	5	5	3	8	9
5	9	3	5	4	3	1
6	3	3	9	10	4	1

Job	Machine sequence					
1	2	0	1	3	5	4
2	1	2	4	5	0	3
3	2	3	5	0	1	4
4	1	0	2	3	4	5
5	2	1	4	5	0	3
6	1	3	5	0	4	2

Table 6.9: Deadlines now assigned to all jobs

Job	Deadline
1	30
2	65
3	60
4	60
5	40
6	55

The obtained schedule for the smallest test instance in Section 6.1, producing the makespan of 57, does not satisfy the deadline constraint and is therefore not feasible. However, Figure 6.6 depicts the production schedule obtained for the smallest test instance, optimized regarding minimizing the makespan and being constrained by the deadlines for each job.

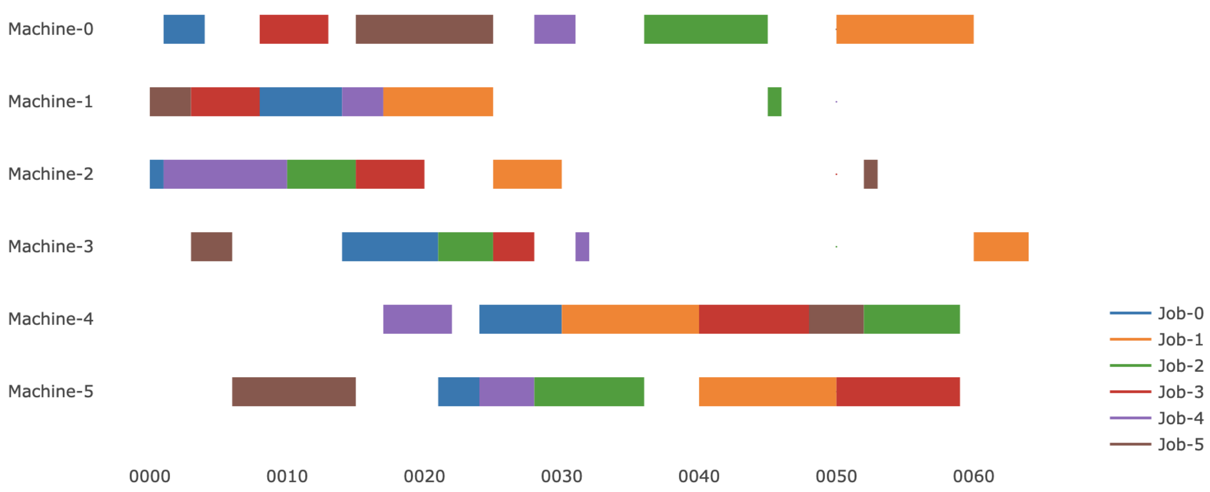

Figure 6.6: Job shop schedule, 6x6 test instance, with objective function to minimize makespan, constrained by deadline

The algorithm carries out an evaluation over which jobs to be scheduled to what time, aiming to satisfy all the deadline values for the jobs. The schedule depicted in Figure 6.6 does indeed satisfy all deadlines. The deadline constraint naturally increases the total makespan compared to the schedule obtained when deadlines were not considered, as can be seen by comparing the two schedules for the *6x6* test instance in Table 6.10.

Table 6.10: Comparison between the obtained schedule of the 6x6 test instance, with and without deadlines

	Previous model	**New model handling deadline**
Deadline	Unavailable	Available
Makespan	57	64

6.3.2 Optimizing the schedule based on makespan and constrained by deadlines for the *20x10* test instance

The same *20x10* test instance used for the model in Section 6.1 is used for demonstrating the optimization model using the genetic algorithm with regards to the deadline. In addition to a specified sequence of machines and processing times as depicted in Table 6.11, there has been added a value for the deadline, depicted in Table 6.12 for the test instance. The specific deadline values are yet again selected such that the obtained schedule for the *20x10* test instance in Section 6.1 would fail to satisfy the deadlines. This forces the algorithm to produce a new schedule, striving to avoid exceeding the jobs' deadline.

Table 6.11: Original input data for the *20x10* test instance

Job	**Processing times in sequence**									
1	52	26	71	16	34	21	95	21	53	55
2	55	98	39	79	12	77	77	66	31	42
3	37	92	64	54	19	43	83	34	79	62
4	87	77	93	69	87	38	24	41	83	60
5	98	25	75	77	49	17	79	44	43	96
6	7	61	95	35	10	35	28	76	95	9
7	59	43	46	28	52	16	59	91	50	27
8	9	43	14	71	20	54	41	87	45	39
9	28	66	78	37	42	26	33	89	33	8
10	96	27	78	84	94	69	74	81	45	69
11	24	32	25	17	87	81	76	18	31	20
12	90	28	72	86	23	99	76	97	45	58
13	17	98	48	46	27	67	62	42	48	27
14	80	50	19	98	28	50	94	63	12	80
15	72	75	61	79	37	50	14	55	18	41
16	96	14	57	47	65	75	79	71	60	22
17	31	47	58	32	44	58	34	33	69	51
18	44	40	17	62	66	15	29	38	8	97
19	58	50	63	87	57	21	57	32	39	20
20	85	84	56	61	15	70	30	90	67	20

Job	**Machine sequence**									
1	8	7	6	9	2	1	5	4	0	3
2	4	5	3	9	0	8	6	7	2	1
3	5	4	2	6	1	7	0	3	9	8
4	1	5	0	3	2	7	8	6	9	4
5	2	5	6	9	1	3	8	0	7	4
6	1	4	0	2	9	8	5	3	7	6
7	5	9	0	4	6	3	2	1	8	7
8	5	9	8	7	4	6	3	0	1	2
9	1	8	0	2	9	3	5	6	4	7
10	4	3	6	5	2	8	1	9	7	0
11	4	7	9	2	3	8	5	6	1	0
12	8	5	1	7	2	3	6	9	4	0
13	2	4	3	1	8	6	7	0	9	5
14	0	8	3	7	5	2	4	6	1	9
15	9	0	4	8	6	2	5	3	7	1
16	3	2	5	0	7	4	8	1	6	9
17	1	7	8	3	4	5	6	0	2	9
18	1	7	2	0	8	6	3	9	5	4
19	2	3	4	9	0	6	7	8	1	5
20	1	0	5	3	9	7	8	2	6	4

Table 6.12: Deadlines now assigned to all jobs

Job	**Deadline**
0	900
1	900
2	900
3	900
4	900
5	1400
6	1400
7	1400
8	1400
9	1400
10	1400
11	1400
12	1400
13	1400
14	1400
15	1400
16	1400
17	1400
18	1400
19	1400

Figure 6.7 depicts the production schedule obtained with the *20x10* test instance, aiming to minimize the makespan and satisfy the deadlines for each job.

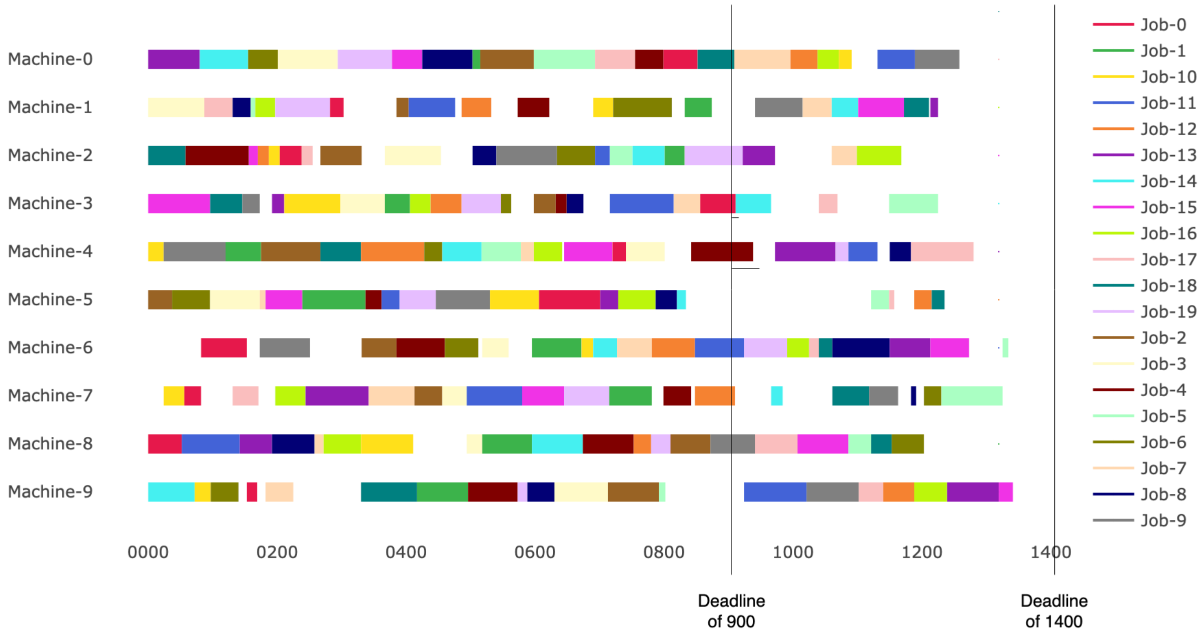

Figure 6.7: Job shop schedule, 20x10 test instance, with objective function to minimize makespan, constrained by deadline, but exceeding deadline

By regarding Figure 6.7 and Table 6.13, it can be seen that two jobs slightly exceed the deadline, Job 0 and Job 4, underlined to the right of the deadline of 900. In fear of being unlucky with the algorithm run, the input data was run multiple times with different parameter tuning, as described in Section 3.4.6, but neither the number of jobs delayed or the amount of time they were delayed, did decrease. It can therefore be argued that there does not exist a solution which satisfies the constraints for the specific input data, consisting of multiple jobs, with operations to be completed on multiple machines in sequence, with deadlines for each job.

Table 6.13: Jobs exceeding the deadline in the 20x10 test instance

Jobs	Deadline	Completion time
Job 0	900	911
Job 4	900	938

The occurrence of two jobs exceeding the deadline is worrying, as stated in Section 3.2, thus such a schedule, estimating delays already before production has started, should not be carried out. It is evident that the existence of both jobs results in both jobs being delayed, as they utilize the same machines, delaying each other. In an attempt to produce a feasible schedule, the deadline of Job 4 was extended to 1400, while the deadline of Job 0 was kept at 900, and the algorithm was run once again. The extension of Job 4 simulates an agreement with the customer of the specific job, proposing a later delivery than originally planned.

Figure 6.8 depicts the new schedule, considering the newly established deadlines. The total makespan is affected, slightly decreased from 1341 to 1335 when Job 4 had its deadline extended from 900 to 1400. The decrease of total makespan due to extended deadline makes sense as the operations of Job 4 now have a broader range of time to be executed, granting additional solutions.

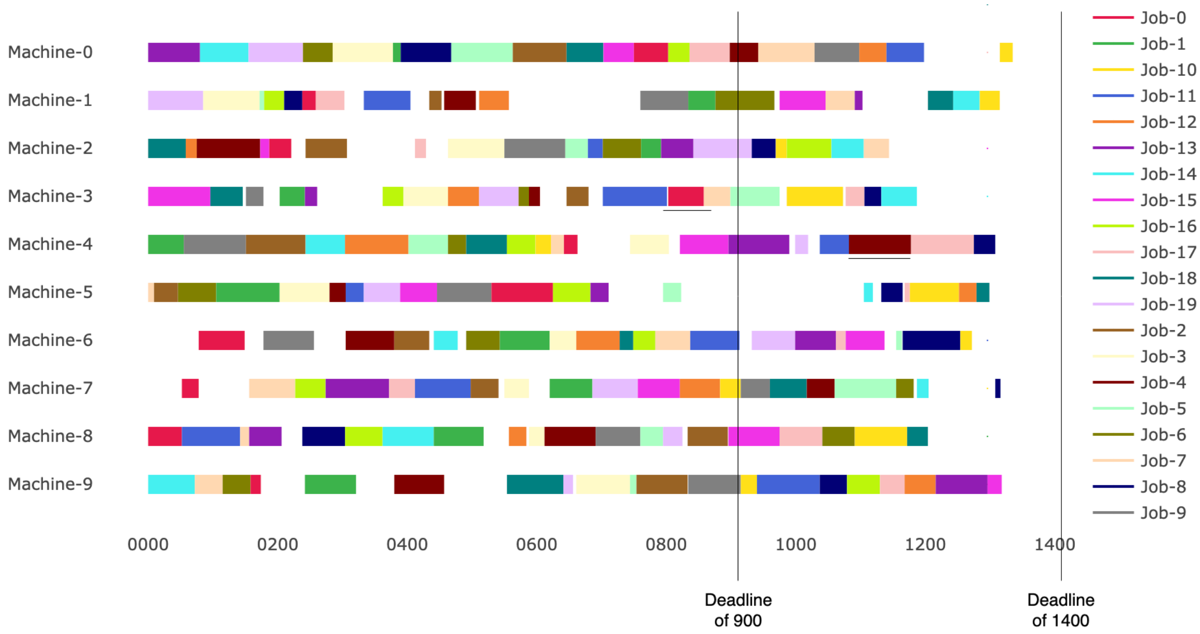

Figure 6.8: Job shop schedule, 20x10 test instance, with objective function to minimize makespan, constrained by deadline, satisfying deadline

The schedule in Figure 6.8 does indeed satisfy all deadlines. The deadline constraint naturally increases the total makespan compared to the schedule obtained when deadlines were not considered, as can be seen by comparing the two schedules for the *20x10* test instance in Table 6.14.

Table 6.14: Comparison between the obtained schedule of the 20x10 test instance, with and without deadlines

	Previous model	New model handling deadline
Deadline	Unavailable	Available
Makespan	1218	1335

As can be seen from Figure 6.8, Job 0, underlined to the left of the deadline of 900, now makes its way within the deadline time. Furthermore, Job 4 is postponed such that it is no longer obstructing Job 0. All jobs now satisfy the deadline constraint. The completion times of once delayed jobs are depicted in Table 6.15.

Table 6.15: Jobs no longer exceeding the deadline in the test instance 20x10

Jobs	New deadline	Completion time
Job 0	900	858
Job 4	1400	1178

As the objective function aims to optimize the schedule based on minimizing jobs exceeding their deadline, in addition to makespan in general, it was essential to penalize schedules if the jobs' completion date exceeded the jobs' deadline. In order to discard solutions which contained delayed jobs, a penalty value was added to the makespan if jobs indeed were delayed, illustrating the occurrence of a job's completion time exceeding its deadline. For each iteration, all jobs' completion times are compared to their deadlines. The equation below depicts the penalty function, where job i's completion time is denoted by J_{ic}, job i's deadline is denoted by J_{id}, and the number of jobs is denoted by n.

$$penalty = \sum_{i=0}^{n} max\{\ 0,\ J_{ic} - J_{id}\}$$

Figure 6.9 depicts the effect of the penalty function, penalizing the makespan value as jobs exceed their deadline by the time each job exceeds its deadline. Penalties for all jobs are added to the makespan value. Initially, the total value of makespan and penalties exceeds 5000, but for each iteration, the number of jobs exceeding their deadline decreases, thus decreasing the sum of the makespan and the penalty. At last a solution is obtained with no jobs exceeding their deadline, thus no penalty, with the value depicting only the makespan.

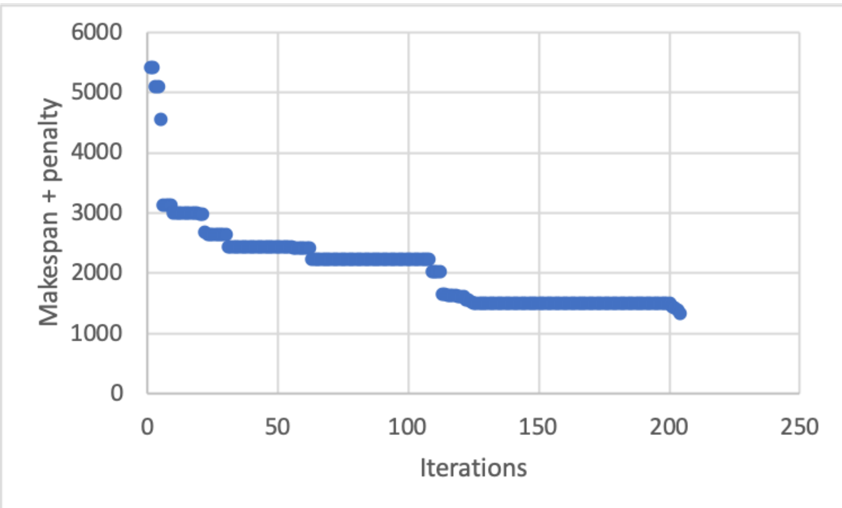

Figure 6.9: The obtained makespan after each iteration, converging to a value of 1335 after 205 iterations

Figure 6.10 depicts the two jobs which once were delayed. Specifically, it can be seen that the operation of Job 4 on Machine 0 delays Job 0 on Machine 0, eventually resulting in the last operation of Job 0, namely the one on Machine 3, to exceed its deadline.

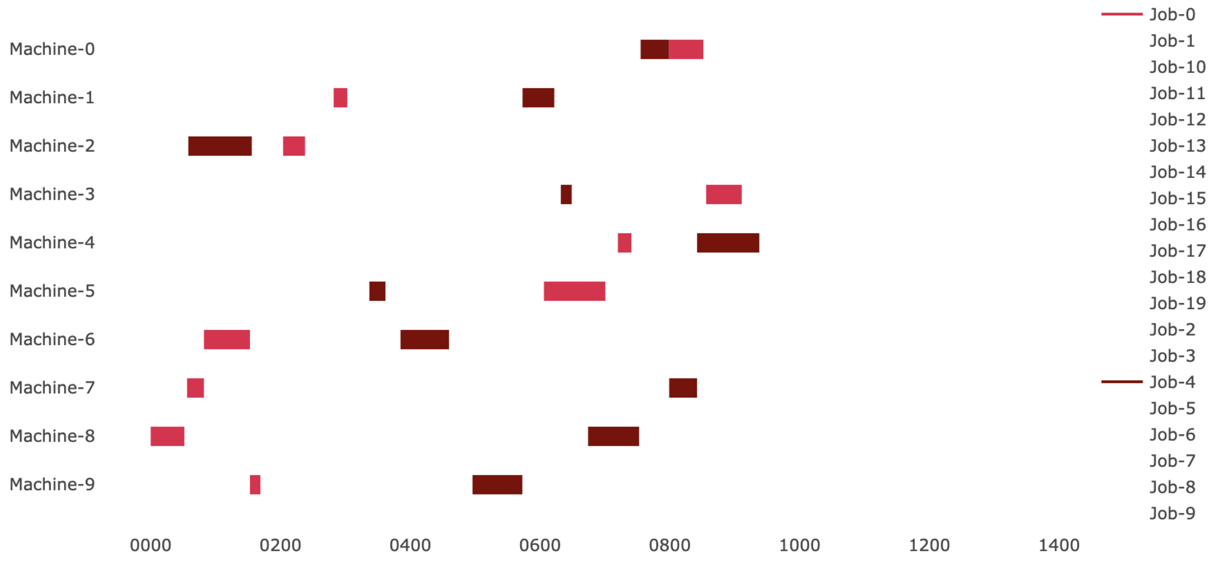

Figure 6.10: Isolated delayed jobs

Figure 6.11 depicts the improved scheduling of Job 0 and Job 4 after extending the deadline of Job 4. Job 0 is no longer delayed on Machine 0, as the two jobs can be scheduled more comfortably due to the extended deadline of Job 4.

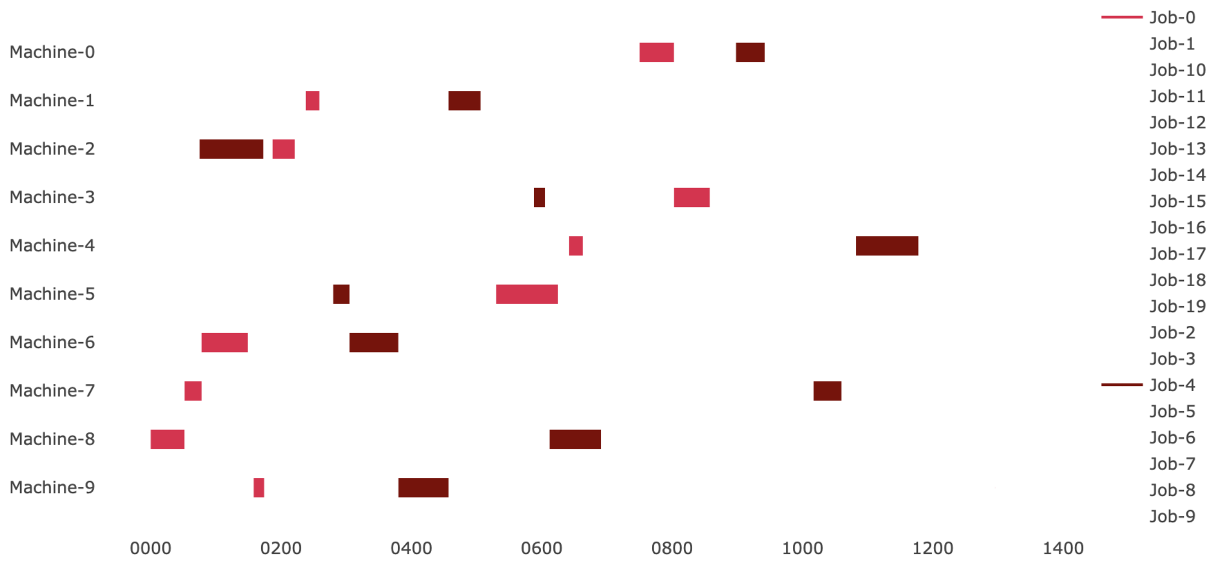

Figure 6.11: Isolated jobs no longer delayed

6.4 Main findings

Three unique models have been developed, optimizing the input data based on different parameters and providing various abilities, as stated in Table 6.16.

Table 6.16: Parameters which are optimized and planning capability in the various developed models

	Initial model	Tailored model for case company	Tailored model for industry
Makespan	X	X	X
Deadline		X	X
Precedence among operations	X		X
Late start		X	
Rescheduling	X	X	X

The initial model was highly successful in optimizing multiple jobs, consisting of several operations to be scheduled on several machines, in precedence, based on the makespan. Due to the strengths which this model depicted, it was further developed to handle data from the case company, data characterized by the existence of deadlines. The developed optimization model for the case company was successful in scheduling operations such that deadlines were considered, in addition to minimizing the makespan. Additionally, scheduling operations with a late start approach has been enabled, such that inventory holding costs are reduced. As the case company data lacked precedence among operations, jobs were executed in parallel, in great contrast to assumptions for the initial model. Therefore, yet another model was developed which utilized the same input data as the initial model, but with deadlines assigned to each job, enabling effective scheduling based on makespan, precedence among operations, and satisfying deadlines.

As the initial model and the final model produce schedules based on the same input data, but with only the latter model considering deadlines, explicitly comparing the two can be done. As can be seen in Section 6.3, when optimizing on deadlines in addition to makespan, the makespan value increases compared to when optimizing solely based on the makespan. This result is expected, as the act of satisfying deadlines results in certain jobs being sequenced at an earlier stage than previously required.

Chapter 7

Discussion

Chapter 6 depicted results from the developed optimization models, obtained by granting input data to the genetic algorithm and performing genetic operators to them. The various production schedules present the solution to various input data, characterized by their corresponding objective functions. Each objective function has its strengths, tailoring the schedule thereafter, and the establishment of objective functions is highly dependant on the specific supply chain. A discussion follows, relating the different optimization models to findings from the literature study which is presented in Chapter 3, and depicting strengths and weaknesses of the various developed optimization models.

7.1 Discussion regarding results of data from benchmark tests

The results obtained in Section 6.1 depict schedules which are attempted to be optimized regarding only minimizing makespan. The objective of minimizing the makespan of jobs has formed the basis for years of researching, (Yin et al., 2011), (Zhang et al., 2017), (Graham, 1966), discovering effective ways to optimize the sequence of operations such that the time from the first operation takes place, until the last operation is completed, is minimized. This objective function can be argued to excel when one batch of orders needs to be completed in the shortest amount of time with deadlines not explicitly considered. (Yin et al., 2011) proposed a genetic algorithm which excels in minimizing the makespan in job shop scheduling. An area of application for this objective function can be argued to be the completion of rush orders, namely products which have priority over already planned production. Supplying the developed optimization model with input data reflecting the products contained in the rush order will produce a schedule minimizing the lead time for the products consisted in the rush order. However, in events where deadlines are an essential trait to an order, optimizing solely based on the makespan is a terrible choice. A schedule optimized solely on makespan will exceed deadlines if the deadline values are somewhat strict, as depicted in Section 6.3.2.

It is worth noting the significant search space of possible solutions of the *20x10* test instance, of which one of the obtained solutions was the result. As each machine schedules 20 operations, each machine has a total of 20! $(20x19x18...x2x1)$ unique operation sequences to settle with. Thus, the schedule which tied with the best-known solution was obtained among a number of 2.43×10^{18} possible solutions, and this is considering only one machine. This finding correlates with the theory of genetic algorithms as proposed in Section 3.4.2, proven to be highly suitable for solving large optimization problems. This, in turn, corresponds with literature regarding the complexity of job shop problems, as presented in Chapter 3 among papers such as (Garey et al., 1976) and (Pishvaee et al., 2011). The obtained results share similarities to the results which were presented in the research paper (Yin et al., 2011), which was the main source of inspiration regarding the development of this model.

By developing the optimization model, based on the genetic algorithm, followed by testing it with benchmark data as described in Section 6.1, the second research objective which was to "utilize an effective algorithm (and adapt the algorithm) to handle job shop scheduling with characteristics of HVLV environments" was approached.

7.2 Discussion regarding results of data from case company

The results obtained in Section 6.2 depict schedules reflecting data from the case company, optimized by two different objective functions. The optimization model which was used to produce the results in Section 6.1 has been further developed in order to consider jobs' deadlines, as deadlines are an essential trait of the case company's customer orders, aiming to increase customer satisfaction as discussed in Section 3.2. The most significant difference between the scheduling of input data from benchmark tests and the company is that the case company's jobs' operations do not require that the previous job's operation is completed before the next can commence. This enables a parallel production sequence among operations of the same job. This discovery was unexpected, as the initially developed model was oriented around precedence of operations. However, the data was susceptible to adaption in order to serve as input to the optimization model. The model was tailored to handle characteristics of the case company, such as deadlines and lack of precedence among operations.

Figure 6.4 depicts the produced schedule based on input data shown in Table 6.7, and is optimized regarding the jobs' completion times not exceeding the jobs' deadlines, and secondly aiming to minimize the makespan. The optimization model does indeed schedule all jobs such that the deadlines are not exceeded, although the sequence of operations on the machines does not reflect the sequence of ascending deadlines. Ideally, once all operations of a job are complete, further manufacturing or assembly of the collective components can take place. The deadline of all jobs should therefore be reflected in the sequence of operations in the schedule.

Figure 6.5 depicts the produced schedule based on the same input data shown in Table 6.7; however, the objective function is no longer based on considering deadlines and makespan, but rather completing the operations as close to the deadline as possible, without exceeding any deadlines. As can be seen from the schedule in Figure 6.5, Job 1's operation on Machine 13 is complete before the remaining operations of Job 1, as the following operation on the same machine requires additional processing time. The schedule does indeed satisfy all deadlines and does postpone the completion time for each job's operations close to the deadline. Postponing each order to the extent as depicted in the schedule leaves the jobs vulnerable to exceeding the deadlines, as any disruption to an operation would result in a delay which is challenging to catch up to. Some additional slack before the deadline is therefore advised.

Advantages and disadvantages may be drawn from both schedules. The first objective function aims to manufacture all jobs in the least amount of time in addition to considering deadlines, resulting in intensive production on the shop floor for a shorter period of time and increased inventory holding due to products being complete at an early stage. However, the slack between each job's operation and its deadline is comfortable, which can be argued to reduce the probability of jobs exceeding their deadlines. Additionally, machine capacity is freed, enabling additional manufacturing of new products, such as rush orders or spare parts.

The second objective function aims to level production, and complete all jobs close to their deadline, resulting in reduced inventory holding and less intensive production on the shop floor. However, the probability of exceeding the deadline increases as the planned completion time approaches the deadline.

A trade-off arises when evaluating both objective functions, one resulting in a greater probability of delivering in time, while the other values the decreased cost of inventory holding more than early completion times. Determining which objective function that is most fitting for a given supply chain requires a cost-benefit analysis, encompassing possible savings of reduced inventory holding compared to the cost of penalties and frequency of completion times exceeding the deadline. Regarding this thesis' case company, with characteristics of today's planning with production constantly being three weeks late, an objective function of minimizing the makespan is argued to assist better with tackling the challenge than by decreasing inventory costs. The additional slack which this objective function entails may be sufficient to avoid exceeding deadlines. However, fixing the cause of the constant delay of three weeks by scheduling based on current capacities rather than utilizing previous plans, is proposed as an even better initiative.

Common for both objective functions, namely the act of explicitly considering deadlines, is in agreement with literature regarding industries operating in the same production environment, as stated in Section 3.1 by (Adrodegari et al., 2015), (Singh, 2006) and (Tamilarasi et al., 2010). Furthermore, additional criteria, such as makespan and inventory holdings are also implemented based on findings in the same literature. As a further development of the model resulted in company data being subjected to optimization, the second research objective of utilizing and *adapting* the algorithm to handle job shop scheduling with characteristics of HVLV environments has been accomplished.

7.3 Discussion regarding results obtained for industries with precedence among jobs' operations

The results obtained in Section 6.3 depict schedules reflecting input data similar to the input data used in Section 6.1, however, similar as to the previous section, a value has been added to represent deadlines for each job. The optimization model which optimized solely based on makespan has been further developed to consider deadlines, by adapting the objective function, adding a penalty value for each time a job exceeds its deadline. Figure 6.8 in Section 6.3 depicts a schedule which satisfies strict deadlines to an optimization problem which is argued in Section 3.2.2 to be incredibly hard to schedule without complex mathematical logic. The rescheduling capability was depicted in Section 6.3 by extending one out of two delayed jobs' deadline and thus producing a feasible schedule which satisfies the deadline for all jobs. Worth noting regarding this practice is that several jobs may be part of one shipment, thus delaying one job in a shipment would delay the remaining complete jobs. Selecting appropriate jobs to establish extended deadlines for is therefore essential. The ability to dynamically reschedule plans according to changes is proposed as one of APS systems' greater strengths, argued for in literature (Ivert, 2009), (Genin et al., 2007).

As this developed model considers multiple jobs, multiple operations in a specific sequence, deadlines considered and satisfied, in addition to the ability to frequently reschedule, this result is argued to handle the most complex data among the obtained schedules. Also, the production environment in question is argued to be the most fitting area of application for the developed optimization model, as the optimization model excels with the existence of precedence among operations.

7.4 The developed models compared to the core model of APS systems

Due to the complexity which a job shop environment places on production scheduling processes, research argues that this environment is the most suitable for implementing advanced planning and scheduling (APS) systems, (Vollmann, 2005), (Jonsson and Mattsson, 2009). Studies regarding implementations of APS systems have been done, (Wiers, 2002), (Zoryk-Schalla et al., 2004), (Stadtler and Kilger, 2002), and the experiences regarding the success have been documented. Benefits regarding all modules of APS systems have been discussed in Chapter 3.3, and it has been highlighted that some APS vendors offer specific modules to cope with specific tasks. The module of interest is the production planning and scheduling module, which encompasses the topics in this thesis' problem formulation. A common challenge regarding the implementation of this module in supply chains is the establishment of assumptions that do not correspond with reality, for example, the occurrence of machine breakdowns or end products not meeting the expected quality metrics. In order to cope with these challenges, the abilities to reschedule frequently and introduce rush orders are enabled, making it possible to produce new schedules that more accurately reflect reality.

The developed optimization models discussed in the sections above are limited to handling deterministic input data, such as processing times, the number of jobs, the number of operations, the number of machines, and the jobs' deadlines, without considering uncertainties or inaccuracies. There are additional parameters which are considered in APS systems' optimization models such as machine efficiency, real-time information, and seamless integration among other APS modules as described in Section 3.4. In other words, it is essential to integrate any optimization model into the supply chain's systems such as an ERP system which provides input, in addition to alternative APS modules, in order to provide accurate data. The developed optimization model is not integrated into a bigger system, and therefore provides results based on an idealized production environment.

APS systems have been defined as "a computer program that uses advanced mathematical algorithms or logic to perform optimization", (Ivert, 2012), a characteristic found in the contribution of the developed optimization models. Although the developed models are independent optimization models, they depict advanced scheduling strengths, strengths which can be argued to become further prevalent when implemented among additional systems.

7.5 Summary of the discussion

APS systems provide benefits concerning decision support, planning efficiency, constraint-based planning, and optimization of parameters. The developed optimization models have aimed to depict planning capabilities which are unavailable in traditional planning systems. Literature has identified APS systems' functionality which by far outperforms the planning and scheduling functionality of ERP systems, (Hvolby and Steger-Jensen, 2010). The optimization models have been developed based on assumptions in their respective production environments, in the context of APS systems' production planning and scheduling module.

Which optimization model, and thus which objective function, that is the most fitting for developing schedules, depends on the specific supply chain's objectives and production environment. If a supply chain aims to complete jobs in the shortest time possible, in order to free capacity for future jobs, the initially developed model excels. However, if a supply chain is highly dependant on meeting deadlines, while manufacturing products which do not require precedence among operations, the second optimization model is suitable. This model also enables a late start of operations and optimizing based on makespan. Furthermore, if a supply chain manufactures jobs with the existence of strict precedence among operations, highly dependant on meeting deadlines, and aiming to minimize the makespan, the third optimization model excels in producing effective and feasible solutions.

All developed optimization models grant the ability to frequently reschedule when new information regarding disruptions on the shop floor or rush orders becomes available, although without an automatic rescheduling which an APS system may provide. However, the developed optimization models provide advanced planning capabilities, enabled by advanced mathematical algorithms and logic to perform optimization, corresponding to the definition of APS systems' computer program, (Ivert, 2012).

7.6 Limitations

Limitations of this study are related to the scope and the research methods which have been applied.

As only four of the six steps in the operations research methodology have been carried out in this thesis, presented in Table 2.1, extensive feedback by management has not been obtained, feedback which would be beneficial to further adapt the optimization models.

Furthermore, the established assumptions which formed the basis of the logic for the developed optimization models are based on deterministic values. Simulations of tackling disruptions on the shop floor in addition to coping with planned scheduling exceeding deadlines have been done, but coping with a collection of several disruptions have not been carried through.

The production planning and scheduling have been identified as effective and efficient in producing optimal schedules when using benchmark tests, but when deadlines were manually assigned to the benchmark test's jobs to depict the capability to handle deadlines, there were no available results to compare to. The produced schedules depict completion times which satisfy strict deadlines, however, describing the obtained schedules in Section 6.3 as optimal is therefore not possible due to lack of data to compare with. In Chapter 8, recommendations for further work are presented based on the limitations of this study.

Chapter 8

Conclusion

The problem formulation in this study was founded by the complexity of production planning and scheduling in high-variety, low-volume production environments. The scope of the thesis is limited to scheduling jobs in an effective manner with HVLV characteristics, in addition to focusing on the benefits of the production planning and scheduling module of APS systems.

8.1 Conclusion

Three smaller objectives were established in order to approach the main objective, which was to develop an effective, efficient and generic optimization model based on a genetic algorithm that performs job shop scheduling in regards to minimizing the plan's total makespan and considering the orders' delivery dates. The first objective was to map and analyze the characteristics of HVLV environments and challenges of job shop scheduling. The findings which shed light on production planning and scheduling, scheduling theory, planning systems and, artificial intelligence formed the basis for the model development and identified planning systems which aim to tackle the documented challenges in this production environment. The findings of the literature study have been presented in Chapter 3 and established assumptions for the models drawn from literature are proposed in Chapter 4. Data collected from the case company was utilized in order to grasp an understanding of challenges in a specific supply chain, which also influenced the assumptions to the models, which is presented in Chapter 5. Hence, the first research objective is met. Furthermore, the obtained results and their corresponding input data are depicted in Chapter 6. Key takeaways from the results are immediately described in the same chapter; however, the results are more thoroughly discussed in Chapter 7.

The three optimization models presented in Chapter 6 are iteratively developed, all based on the foraging behavior of bees. Genetic algorithms have proven to be effective in obtaining solutions to large optimization problems, as described in Section 3.4.2. The first model, which optimizes solely based on the total makespan of all jobs, was developed based on a research paper, (Yin et al., 2011), which presented the logic of the genetic algorithm. The model was proved to be effective by being tested on benchmark test data. Furthermore, a second model was developed, building on the already developed one. The first model did not consider deadlines, a characteristic highly prevalent in the case company's environment. However, the case company's production environment required no precedence among a job's operations, resulting in the operations to be manufactured in parallel.

Further development of the model encompassed these new characteristics and produced schedules based on two objective functions, one aiming to minimize the total makespan in addition to satisfying deadlines, while the other aimed to schedule such that the jobs' planned completion time approached the deadline as closely as possible in order to reduce inventory holding costs, while satisfying deadlines.

Lastly, as the optimization of deadlines was a success, it was argued that adding deadlines to the complex production routing, which the jobs in Section 6.1 consisted of, could be beneficial in order to develop an optimization model suitable for companies with those characteristics. Optimizing in regards to delivery dates in addition to makespan was successful. The main contribution regarding the model development is indeed this final model which encompasses jobs with a significant degree of precedence among operations, on multiple machines, with deadlines considered. Hence, the second research objective, utilizing an effective genetic algorithm and adapting the algorithm to handle job shop scheduling with characteristics of HVLV environments, is met.

Advanced planning and scheduling (APS) systems have been proposed in the literature as highly suitable for coping with challenges in job shop scheduling environments, as stated in Section 3.3. Capabilities far exceeding those of enterprise resource planning (ERP) systems and Microsoft Excel have been documented, and are regarded as essential in order to cope with the increasingly complex characteristics of this production environment. The developed optimization models have aimed to depict strengths of the core model of such a system, without being integrated among other systems, which an APS system requires. Hence, the third research objective is met.

8.2 Recommendations for further work

The scope of this study has been limited to operational planning of industries operating with the scheduling of jobs, within the HVLV environment. All three developed optimization models may be subject to further development. The sequence of operations could be further optimized based on reducing tool changes, as extensive changeovers are time-taking. A recommendation for future work is therefore to increase the number of constraints and parameters that the optimization models consider, such that the results more accurately reflect the real world.

Identifying test data of job shop scheduling with assigned deadlines could assist in depicting the benefits of scheduling with a genetic algorithm such as the artificial bee colony by comparing with other approaches.

References

Adrodegari, F., Bacchetti, A., Pinto, R., Pirola, F., and Zanardini, M. (2015). Engineer-to-order (eto) production planning and control: An empirical framework for machinery-building companies. *Production Planning & Control*, 26(11):910–932.

Aldowaisan, T. and Allahverdi, A. (2003). New heuristics for no-wait flowshops to minimize makespan. *Computers & Operations Research*, 30(8):1219–1231.

Amaro, G., Hendry, L., and Kingsman, B. (1999). Competitive advantage, customisation and a new taxonomy for non make-to-stock companies. *International Journal of Operations & Production Management*, 19(4):349–371.

Andresen, M., BräSel, H., MöRig, M., Tusch, J., Werner, F., and Willenius, P. (2008). Simulated annealing and genetic algorithms for minimizing mean flow time in an open shop. *Mathematical and Computer Modelling*, 48(7-8):1279–1293.

Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

Bierwirth, C. and Mattfeld, D. C. (1999). Production scheduling and rescheduling with genetic algorithms. *Evolutionary computation*, 7(1):1–17.

Bonabeau, E., Marco, D. d. R. D. F., Dorigo, M., Theraulaz, G., et al. (1999). *Swarm intelligence: from natural to artificial systems*. Number 1. Oxford university press.

Boonsothonsatit, K. (2016). Supply chain causal linkage-based strategic map design. *Journal of Advanced Management Science*.

Boushaala, A., Shouman, M., and Esheem, S. (2012). Some heuristic rules for job shop scheduling problem. *in Engineering Journal of Alexandria, EJA*.

Bringsjord, S. (1998). Chess is too easy. *Technology Review*, 101(2):23–28.

Buetfering, B., van der Stoel, A., Sriram, P. K., Van Landeghem, H., Ickerott, I., and Wiegel, V. (2016). Lean in high variety, low volume production environments- a literature review and maturity model.

Chua, T. J., Wang, F. Y., Yan, W. J., and Cai, T. X. (2011). Rough-cut capacity planning with production constraints and dynamic bottleneck considerations. US Patent 7,925,365.

Croom, S. (2010). Introduction to research methodology in operations management. In *Researching operations management*, pages 56–97. Routledge.

Dahal, K., Tan, K. C., and Cowling, P. I. (2007). *Evolutionary scheduling*, volume 49. Springer Science & Business Media.

David, F., Pierreval, H., and Caux, C. (2006). Advanced planning and scheduling systems in aluminium conversion industry. *International Journal of Computer Integrated Manufacturing*, 19(7):705–715.

de Man, J. C. and Strandhagen, J. O. (2018). Spreadsheet application still dominates enterprise resource planning and advanced planning systems. *IFAC-PapersOnLine*, 51(11):1224–1229.

Dilek, S., Çakır, H., and Aydın, M. (2015). Applications of artificial intelligence techniques to combating cyber crimes: A review. *arXiv preprint arXiv:1502.03552*.

Dumitrescu, D., Stoean, C., and Stoean, R. (2007). Genetic chromodynamics for the job shop scheduling problem. In *Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques*, pages 153–160.

Dym, C. (2004). *Principles of mathematical modeling*. Elsevier.

Dym, C. and Ivey, E. (1980). *Mathematical Modeling*. Academic Press, New York.

Gao, W.-f. and Liu, S.-y. (2012). A modified artificial bee colony algorithm. *Computers & Operations Research*, 39(3):687–697.

Garey, M. R., Johnson, D. S., and Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of operations research*, 1(2):117–129.

Genin, P., Thomas, A., and Lamouri, S. (2007). How to manage robust tactical planning with an aps (advanced planning systems). *Journal of Intelligent Manufacturing*, 18(2):209–221.

Ghalayini, A. M., Noble, J. S., and Crowe, T. J. (1997). An integrated dynamic performance measurement system for improving manufacturing competitiveness. *International Journal of production economics*, 48(3):207–225.

Gosling, J. and Naim, M. M. (2009). Engineer-to-order supply chain management: A literature review and research agenda. *International journal of production economics*, 122(2):741–754.

Graham, R. L. (1966). Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 45(9):1563–1581.

Grefenstette, J. J. (2014). *Proceedings of the First International Conference on Genetic Algorithms and their Applications*. Psychology Press.

Haupt, R. L. and Ellen Haupt, S. (2004). Practical genetic algorithms.

Hayes, R. H. and Wheelwright, S. C. (1979). The dynamics of product-process life cycles. *Harvard Business Review*, 57(2):127–136.

Henning, G. P. (2009). Production scheduling in the process industries: current trends, emerging challenges and opportunities. In *Computer Aided Chemical Engineering*, volume 27, pages 23–28. Elsevier.

Hillier, F. S. (2010). *Introduction to Operations Research*. McGraw-Hill, 9 edition.

Hoppe, M. (2007). *Sales and inventory planning with SAP APO*. Galileo Press Boston.

Hvolby, H.-H. and Steger-Jensen, K. (2010). Technical and industrial issues of advanced planning and scheduling (aps) systems. *Computers in Industry*, 61(9):845–851.

Infor (2010). Planning and scheduling in the 21st century.

Ivert, L. K. (2009). *Advanced planning and scheduling systems in manufacturing planning processes*. Chalmers University of Technology.

Ivert, L. K. (2012). *Use of Advanced Planning and Scheduling (APS) systems to support manufacturing planning and control processes*. Chalmers University of Technology.

Jonsson, P. and Backstrom, C. (1995). Incremental planning. In *New Trends in AI Planning: Proc. 3rd European workshop on planning. IOS*, pages 79–90. IOS Press.

Jonsson, P. and Mattsson, S.-A. (2009). Manufacturing planning and control.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer ....

Karaboga, D. and Basturk, B. (2008). On the performance of artificial bee colony (abc) algorithm. *Applied soft computing*, 8(1):687–697.

Karafotias, G., Hoogendoorn, M., and Eiben, Á. E. (2015). Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 19(2):167–187.

Kristianto, Y., Ajmal, M. M., and Helo, P. (2011). Advanced planning and scheduling with collaboration processes in agile supply and demand networks. *Business Process Management Journal*, 17(1):107–126.

L BERG, B. (2001). Qualitative research methods for the social sciences.

Lupeikiene, A., Dzemyda, G., Kiss, F., and Caplinskas, A. (2014). Advanced planning and scheduling systems: modeling and implementation challenges. *Informatica*, 25(4):581–616.

Lütke Entrup, M. (2005). Advanced planning in fresh food industries: integrating shelf life into production planning. *Physica, Heidelberg*.

Mao, W. (1995). Multi-operation multi-machine scheduling. In *HPCN Europe*.

Mattfeld, D. C. (2012). Test instances scheduling.

Mbaskool (2019). Engineer-to-order.

Meyr, H., Wagner, M., and Rohde, J. (2008). Structure of advanced planning systems. In *Supply chain management and advanced planning*, pages 109–115. Springer.

Moon, C., Kim, J., and Gen, M. (2004). Advanced planning and scheduling based on precedence and resource constraints for e-plant chains. *International Journal of Production Research*, 42(15):2941–2955.

Naden, J. (2000). Have a successful. *IIE Solutions*, 32(10):46–46.

Nishioka, Y. (2005). Advanced planning and scheduling (aps) conceptual definition and implementation. *White Paper, PSLX Consortium.*

Patel, V. L., Shortliffe, E. H., Stefanelli, M., Szolovits, P., Berthold, M. R., Bellazzi, R., and Abu-Hanna, A. (2009). The coming of age of artificial intelligence in medicine. *Artificial intelligence in medicine*, 46(1):5–17.

Pishvaee, M. S., Rabbani, M., and Torabi, S. A. (2011). A robust optimization approach to closed-loop supply chain network design under uncertainty. *Applied Mathematical Modelling*, 35(2):637–649.

Portioli-Staudacher, A. and Tantardini, M. (2012). A lean-based orr system for non-repetitive manufacturing. *International Journal of Production Research*, 50(12):3257–3273.

Rajesh, R. (2018). Pseudo resilient supply chains: concept, traits, and practices. *Journal of Risk Research*, 21(10):1264–1286.

Röck, H. (1984). The three-machine no-wait flow shop is np-complete. *Journal of the ACM (JACM)*, 31(2):336–345.

Sagadin, J. (2004). *Tipi in vloga študij primerov v pedagoškem raziskovanju*.

Seeley, T. D. (2009). *The wisdom of the hive: the social physiology of honey bee colonies*. Harvard University Press.

Shah, N. K. and Ierapetritou, M. G. (2012). Integrated production planning and scheduling optimization of multisite, multiproduct process industry. *Computers & Chemical Engineering*, 37:214–226.

Sharman, G. (1984). Rediscovery of logistics. *Harvard Business Review*, 5:71–79.

Singh, H. (2006). The importance of customer satisfaction in relation to customer loyalty and retention. *Academy of Marketing Science*, 60(193-225):46.

Slack, N., Chambers, S., and Johnston, R. (2010). *Operations management*. Pearson education.

Sriram, P. K., Alfnes, E., and Arica, E. (2012). A concept for project manufacturing planning and control for engineer-to-order companies. In *IFIP International Conference on Advances in Production Management Systems*, pages 699–706. Springer.

Stadtler, H. (2005). Supply chain management and advanced planning—-basics, overview and challenges. *European journal of operational research*, 163(3):575–588.

Stadtler, H. and Kilger, C. (2002). *Supply chain management and advanced planning*, volume 4. Springer.

Starman, A. B. (2013). The case study as a type of qualitative research. *Journal of Contemporary Educational Studies/Sodobna Pedagogika*, 64(1).

Stavrulaki, E. and Davis, M. (2010). Aligning products with supply chain processes and strategy. *The International Journal of Logistics Management*, 21(1):127–151.

Steger-Jensen, K., Dreyer, H., Hvolby, H.-H., and Standhagen, O. (2014). Towards a green and sustainable manufacturing planning and control paradigm using aps technology. In *IFIP International Conference on Advances in Production Management Systems*, pages 442–449. Springer.

Tamilarasi, A. et al. (2010). An enhanced genetic algorithm with simulated annealing for job-shop scheduling. *International Journal of Engineering, Science and Technology*, 2(1):144–151.

Turbide, D. (1998). Advanced planning and scheduling (aps) systems. *Midrange ERP magazine*, 1.

Umble, E. J., Haft, R. R., and Umble, M. M. (2003). Enterprise resource planning: Implementation procedures and critical success factors. *European journal of operational research*, 146(2):241–257.

Van Eck, M. (2003). Is logistics everything, a research on the use (fullness) of advanced planning and scheduling systems. *BWI paper, Faculty of Sciences, Vrije Universiteit, Amsterdam, April*.

Vieira, G. E., Herrmann, J. W., and Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of scheduling*, 6(1):39–62.

Vollmann, T. E. (2005). *Manufacturing planning and control for supply chain management*.

Webster, J. and Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, pages xiii–xxiii.

Wiers, V. C. (2002). A case study on the integration of aps and erp in a steel processing plant. *Production planning & control*, 13(6):552–560.

Yin, M., Li, X., and Zhou, J. (2011). An efficient job shop scheduling algorithm based on artificial bee colony. *Scientific Research and Essays*, 6(12):2578–2596.

Zhang, J., Ding, G., Zou, Y., Qin, S., and Fu, J. (2017). Review of job shop scheduling research and its new perspectives under industry 4.0. *Journal of Intelligent Manufacturing*, pages 1–22.

Zoryk-Schalla, A. J., Fransoo, J. C., and de Kok, T. G. (2004). Modeling the planning process in advanced planning systems. *Information & Management*, 42(1):75–87.