A capacity-driven due date assignment and workload spreading approach in a make-to-order production environment

Master thesis Industrial Engineering and Management

Nathan Hoogendoorn
August 2023

A capacity-driven due date assignment and workload spreading approach in a make-to-order production environment

Author
Nathan Hoogendoorn
S2152312
MSc Industrial Engineering and Management

TOMRA
Laan van Malkenschoten 80
7333 NP Apeldoorn
Netherlands

University of Twente
Drienerlolaan 5
7522 NB Enschede
Netherlands

Supervisor TOMRA
J. Bakker
TOMRA
Manager production logistics and R&D

Supervisors University of Twente
Dr. Ir. J.M.J. Schutten
Dr. E. Topan

Preface

This thesis has been performed at TOMRA in Apeldoorn and marks the end of my master program Industrial Engineering and Management at the University of Twente. Throughout my studies, I also completed the bachelor program Industrial Engineering and Management besides, I learned a lot and had many great and unforgettable experiences.

I would like to express my gratitude towards TOMRA for giving me the opportunity to do research at their company. It was a great experience to conduct my thesis at a large scale production facility that makes advanced and innovative machinery. I want to thank all of the colleagues at TOMRA for their hospitality and collaboration to be able to conduct the thesis in a pleasant way. Besides, I especially want to thank Jeroen Bakker, my company's supervisor, for the support and guidance during my thesis.

I want to give special thanks to Macro Schutten, as the first supervisor from the University of Twente, for his great support, feedback and suggestions regarding my research. Moreover, I want to thank Engin Topan for his valuable feedback and insights as the second supervisor of my master thesis.

At last, I would like to thank my friends and family for their support and encouragement during this graduation research.

Nathan Hoogendoorn
August 2023

Management summary

The production facility of TOMRA in Apeldoorn is responsible for the production of the backroom of bottle deposit systems. The backroom corresponds to elements located behind collecting devices of empty drinking packages, which are tailor-made and can differ per customer, making the production environment make-to-order (MTO). The facility of TOMRA located in Apeldoorn often experiences production related problems. Among other things, high workloads and inventory shortages make up for these production problems. These problems are mainly caused by the fact that the production capacity is currently not fully considered in production planning activities at TOMRA. More specifically, optimizing due date assignments to incoming orders is of great relevance, where customers currently determine the due dates. As TOMRA does not have detailed capacity overviews, they have little saying in providing feedback on the requested due dates and almost always comply with the customers' requests, resulting in heavy workload fluctuations. Therefore, the main research question is the following:

"How to optimize the combined process of assigning due dates with spreading the workload of incoming orders?

An analysis of the current situation shows that there is need for a tactical level capacity planning at TOMRA. This is because TOMRA is very dependent on their ERP system in the way of operating. This system currently lists necessary production activities per production department based on the earliest due date, resulting in waiting until the very last moment to produce required backroom elements. Moreover, because the ERP system does not consider required and available production capacity for the workload of incoming orders, the current way of operating causes large workload fluctuations.

From the literature, we identify relevant methods that are capable of assigning due dates in a MTO environment. Especially finite loading methods can help in assigning due dates whilst considering available and occupied capacity. However, the finite loading methods do not generate an optimal spread of the existing and incoming workloads, as the purpose of these methods is to solely assign due dates to orders and not the optimal loading of resources. Because of this, we identify capacity planning methods from the literature, which provide us insights into techniques such as imposing tardiness to orders and allowing nonregular capacity usage, to optimize the workload spread of orders.

Using the knowledge from the literature, we develop a finite loading model that is capable of optimizing due date assignments to incoming order requests based on a weekly timeframe. We fit the model to our problem setting with multiple production departments, where customers have a desired due date and no incoming orders are refused. The model considers imposing tardiness to existing orders that have not been completed yet to fit an incoming order in the production capacity. Moreover, we consider the usage of nonregular capacity in this process. In addition, we develop a sample average approximation (SAA) inspired approach that is capable of determining release dates of operations that require one or multiple materials with stochastic external lead times. The SAA method also assesses the feasibility of loading these operations into a certain week to obtain a service level.

The solution approach for solving the model in our setting consists of an approximate method to obtain good quality model solutions in a timely manner by means of heuristics. In total, we consider 6 construction heuristics that can generate initial solutions to our problem, namely FL, CFL, HL, UFL, UCL and UHL. These construction heuristics follow concepts of forward and backward loading to fit the workload of incoming order requests in available production capacity. Moreover, we include two improvement heuristics that can improve on initial solutions, namely an adapted version of steepest descent (ASD) and simulated annealing (SA).

In order to find the best solution approach in our problem setting at TOMRA, we develop an experimental design that evaluates the performances of the approaches based on 4 problem instances. These problem instances contain existing orders from different, but recent moments in time. Each problem instance requires loading of 4 different incoming orders that all request a due date during a peak demand period.

From the experiment results, we conclude that the HL approach shows the overall best performance in terms of objective value. This approach loads the operations within an incoming order either forward or backward, depending on the preferences of the customer and production department. Moreover, both forward loading approaches (FL and UFL) show poor performance related to the service levels as a result of their way of loading the operations. Using only the HL construction heuristic provides already good quality solutions. However, by applying an improvement heuristic, the objective values can be improved. This is done by modifying some loading periods of order operations, to better fit the workload of an incoming order within available production capacity. ASD shows similar performances related to the SA improvement heuristic, however does so in less computation time, making this approach superior with its current parameters in our problem setting.

Finally, we recommend TOMRA to implement the HL approach for loading the workloads of their incoming orders in the future and recommend to apply the ASD improvement heuristic to enhance the quality of solutions in a timely manner.

Table of Contents

1. Introduction

This chapter provides an introduction to the company at issue, namely TOMRA, as well as the purpose of this research. Section 1.1 describes the company TOMRA. Section 1.2 presents the problem statement including the action problem, problem cluster and core problem selection. Section 1.3 introduces the research design, consisting of the research scope as well as research and knowledge questions.

1.1 Company description

At TOMRA, systems are developed and created that help in transforming companies into more profitable and sustainable businesses. A large part of the activities that TOMRA carries out deal with the disposal of drinking packages. TOMRA is market leader in the field of packaging intake by offering solutions to collect, sort and process empty bottles and cans through bottle deposit systems. Every year, approximately 40 billion drinking packages are collected with the use of the systems developed by TOMRA, after which the packages can be recycled. The developed bottle deposit systems, with the corresponding digital tools and services, can provide a user-friendly way for recycling drinking packages.

The facility of TOMRA located in Apeldoorn is responsible for the design and production of the backroom of bottle deposit systems that are being used within supermarket branches all over Europe. The backroom consists of elements located behind the collecting device, which can differ per customer based on specific wishes, intake volume and allocated space. Within the production facility of TOMRA in Apeldoorn, the three main elements of these backroom systems are produced, namely tables, roller belts and conveyors that together are essential parts for sorting and the disposal of drinking packages. These attributes are tailor-made per customer to optimally utilize the available space.

TOMRA produces the backroom elements of the bottle deposit systems once there is demand. The main reason for this way of producing is that the developed backroom systems can vary significantly per supermarket establishment.

1.2 Problem identification

Within this section, we address relevant production related problems that occur at TOMRA. First, the formulation of the main problem that TOMRA currently faces, namely the action problem of the research, takes place. Next, we provide a problem cluster, showing the relationships between the identified problems. Afterwards, we select the core problem stating the main focus of this research.

1.2.1 Action problem

Within the facility of TOMRA in Apeldoorn, problems currently occur within the production department during peak demand periods. Demand for the bottle deposit systems takes place during the entire year. Various times per year, highly increased demand periods occur, causing problems on the workplace to arise, where a tremendous amount of overtime working hours must be made in order to meet the demand. Based on the situation that TOMRA currently encounters, an action problem can be defined.

According to Heerkens and van Winden (2017), an action problem describes the discrepancy between norm and reality as perceived by the problem owner. The problem owner at stake is TOMRA in Apeldoorn. In reality, various problems occur during peak demand periods, such as often working in overtime or not meeting due dates because of inventory shortages. TOMRA wants to prevent these problems in the future, meaning that they want to be able to cope well with the production of the backroom systems during peak demand periods. There is a clear discrepancy between the norm and reality present, making the action problem of this research the following:

> ***"The production facility of TOMRA in Apeldoorn is not able to cope well with the demand of the backroom elements of the bottle*** **deposit** ***systems during peak demand periods"***

1.2.2 Problem cluster

Figure 1.1 shows a problem cluster, consisting of current problems related to the production of the backroom for the bottle deposit systems as well as the relations between the problems. The problem cluster is based on interviews and having verbal contact with the employees of TOMRA within the production and sales departments. Three types of problems can be seen within the problem cluster, namely regular problems, core problems and the previously designated action problem.

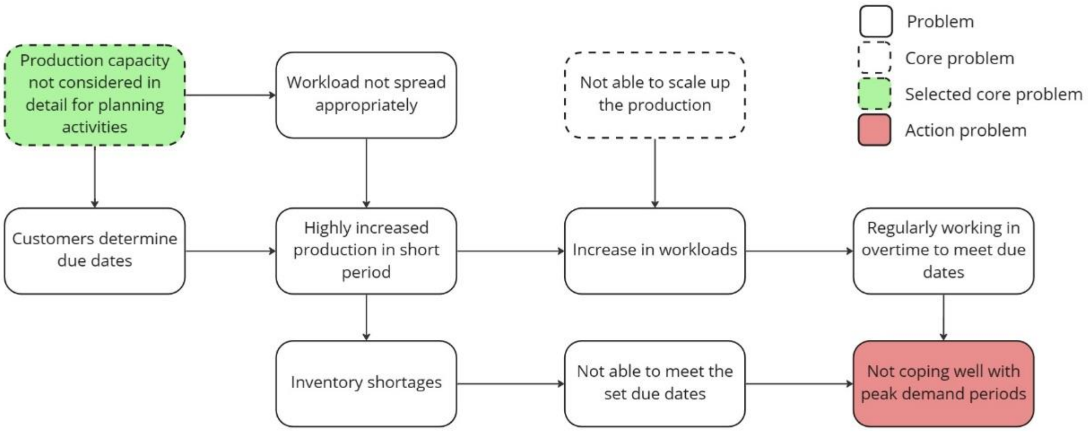

Figure 1.1; Problem cluster

The central action problem is that TOMRA is not able to cope well with highly increased demand periods. These increased demand periods are, for instance, caused by the development of new bottle deposit systems, which can provide new ways for the recycling of drinking packages.

As of right now, the production capacity is not considered in detail for planning related activities. When receiving production orders from supermarkets, the production capacity is not fully taken into account for setting up due dates for these orders. TOMRA lets their customers determine due dates for placed orders caused by a lack of their capacity insights. Subsequently, the production capacity and production workload at the workplace are not distributed accordingly.

During peak demand periods, many (large) production orders occur resulting into great increases within the workloads because of the current due date policy. The workloads also increase, since TOMRA is not able to scale up the production process in a short time span. This is because production expansion in terms of increased space and extra personnel is difficult to achieve in the short term.

Furthermore, inventory shortages regularly take place in several steps of the production process during increased demand periods. These shortages can disrupt and delay the production of the backroom systems and result in not meeting the set due dates. The inventory shortages may also occur due to delayed lead times of suppliers.

1.2.3 Core problem selection

The problem cluster depicted in Figure 1.1 contains multiple core problems. Core problems are found in the roots of the problem cluster, directly causing other problems (Heerkens & van Winden, 2017). The selection of the core problem is a crucial step of this research, as solving the core problem helps to tackle the action problem. Altogether, two different candidate core problems can be identified for TOMRA:

1. Not able to scale up the production
2. Production capacity not considered in detail for planning activities

To be able to select the core problem that is the focus of this research, we investigate both candidate core problems in more detail. Tackling the first candidate core problem, namely not being able to scale up the production, will help TOMRA with coping better with the production during the peak demand periods. However, since TOMRA still experiences various inefficiencies in their current way of planning and assigning due dates, they are aware that tackling the first identified core problem would be a temporary solution. That is why it is convenient to focus on the second identified core problem during this research. Considering the production capacity for planning related activities could prevent the need of suddenly scaling up the production. Summarizing, solving the second candidate core problem would have the largest positive impact in the production of the backroom element in the future. Therefore, the following core problem is selected and is the focus of this research:

> ***"The production capacity is currently not considered in detail for production planning activities"***

1.3 Research design

This section provides the approach of the research to solve the selected core problem. First, we state the research scope, where we outline the focus of the research more specifically. In addition, we formulate research questions, from which the answers are necessary steps in solving the core problem.

1.3.1 Research scope

As the identified core problem relates to including a detailed production capacity overview to produce of the backroom elements for bottle deposit systems, we define a research scope. Especially optimizing both the assignment of appropriate due dates and spreading the workload whilst taking the production capacity into consideration is deemed relevant by TOMRA. In addition, inventory shortages are seen as a drastic problem that can disrupt the production. The purpose of this research is to solve the core problem, where both of these identified important elements are the focus.

Next to the production of the backroom systems, the TOMRA facility in Apeldoorn also accounts for maintenance, cleaning and repair operations to provide service and support to their customers. The scope of this research is related to the main production process of TOMRA, generating most of the output. This main production process takes places at three different departments, which can produce independently of each other to realize a collective output. Figure 1.2 visualizes an overview of the production processes and their relations.

All three main production departments have many items and materials involved to generate the desired output. It is not feasible to consider all items during this research. Because of this, we consider only the items or materials that TOMRA identifies as the most important or the items where shortages often occur.

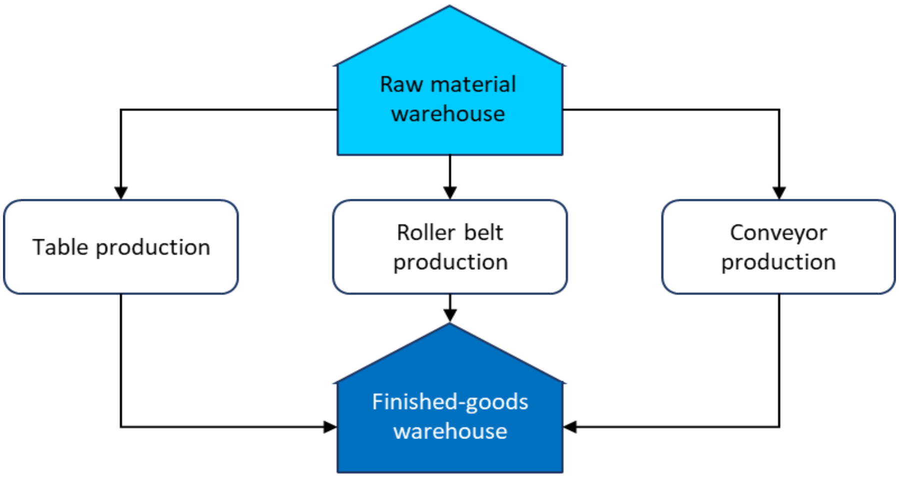

Figure 1.2; Production processes overview

1.3.2 Research questions

To solve the proposed core problem, the following main research question is formulated and indicates the aim of the research:

"How to optimize the combined process of assigning due dates with spreading the workload of incoming orders?

To be able to solve the identified main research question, we split the main research question into multiple more specific research questions.

The first research question aims to analyze the current production process at TOMRA in detail. Information on the necessary steps to complete an order is useful to create a better understanding of the production methods. Besides, it is relevant to understand how production schedules are constructed as well as which software systems are available and in use of supporting the production processes. Chapter 2 provides an answer to the following research question and sub research questions:

1) **How does TOMRA currently operate to manufacture the backroom of the bottle deposit systems?**
   I) What are the production processes and how are these related?
   II) How are production schedules currently constructed?
   III) Which software system(s) is/are used during the production and how are these related?

The next research question aims to translate the problem at hand into related problems within the literature to acquire relevant knowledge. This part is the theoretical framework for the research. We address existing relevant methods for due date generation within the literature. In addition, we research literature regarding the construction of a tactical level planning. In Chapter 3, the literature provides answers to the following research question and sub research questions:

2) **Which solution methods are available within the literature on using production capacity for setting realistic due dates?**
   I) Which methods to generate due dates exist within the literature?
   II) Which methods within the literature are able to generate a tactical level production plan to spread the workload of incoming orders?

The purpose of the following research question is to develop a model that are able to fill in the identified literature gap. This part requires some of the knowledge found within the theoretical framework to develop a model capable of assigning due dates of incoming customer orders and spread the workload. The model should take into account the production capacity in order to generate insights. Besides, since TOMRA desires to diminish the occurrence of inventory shortages during production activities, the model should incorporate an approach for this. We treat these aspects, the research question and sub research questions in Chapter 4.

3) **How can a model be developed that is able to optimize production due date assignments in the setting at TOMRA?**
    I) How can the production capacity be considered?
    II) How can the model diminish the inventory shortages during the production?

Subsequently, the next research question aims to identify solution approaches for the developed model. In Chapter 5, we develop the solution approaches based on findings within the literature with the purpose to eventually find the best approach.

4) **Which solution approaches are capable of solving the model for the combined due date assignment and workload spreading problem at TOMRA?**
    I) How can solutions for the previously developed model be constructed?

At last, we aim to identify the best solution approach from the alternatives developed beforehand. We select the best performing solution approach based on various experimental designs. For this, experiments need to be designed and results should follow. Chapter 6 outlines the following research question and sub research questions:

5) **Which solution approach performs best based on several experiments?**
    I) How to design appropriate experiments?
    II) What are the experimental results per alternative?

2. Current situation

This chapter outlines the current production process and way of operating for the production of the backroom for bottle deposit systems. First, the steps necessary for the table production, roller belt production and conveyor production to come to a desired end product are described in Section 2.1. Section 2.2 provides a description of the IT system in use as well as an overview of the current way of assigning due dates and planning and scheduling activities of the production steps. At last, Section 2.3 presents a conclusion to this chapter.

2.1 Production processes

This section provides an overview of the products made and production activities during the main production processes. Sections 2.1.1, 2.1.2 and 2.1.3 discuss the production activities of the tables, roller belts and conveyors respectively.

2.1.1 Table production

Tables are an essential part of the backroom of bottle deposit systems. A table is situated right behind the bottle deposit device, where a user disposes empty drinking packages. The table is the first element of the backroom and guides drinking packages from the deposit device towards a further destination. At TOMRA, tables are being produced in all kinds of shapes and sizes and differ per supermarket. This makes the tables customizable and is the reason that tables are being produced once an order arrives. The exact specifications of each required table per order are stated and mentioned when an order arrives. Both the shape and type of table affect its production time, for instance the production of a large table consists of a higher workload.

A couple of standard table shapes can be identified, namely a straight table and an L-shaped table. A straight table transports the drinking packages in a straight line and requires the lowest work effort to be made. Figure 2.1 visualizes an example of a straight table with crate rack within a backroom system. Moreover, an L-shaped table has an angle of 90 degrees and can transport the drinking packages towards a different direction. An L-shaped table automatically needs more than one engine, as two belts realize movement along the L-shaped table. This makes workload to produce an L-shaped table to be higher.

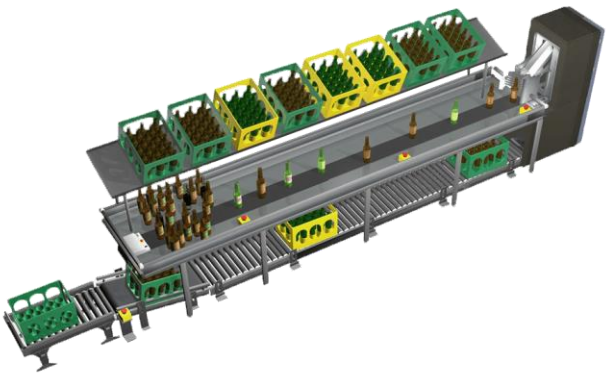

Figure 2.1; Example table in backroom setting

TOMRA offers possibilities for two different types of backroom support systems. These systems are named EasyPac and Multipac and can be viewed in Figures 2.2 and 2.3 respectively. An EasyPac machine is able to automatically sort PET bottles and cans and makes them compact. A MultiPac machine is an extension of the EasyPac system and can do this for two bottle collecting devices at once. The facility of TOMRA in Apeldoorn does not produce these machines. Both the EasyPac and MultiPac machines are produced at the headquarters in Norway. The EasyPac and MultiPac machines can either be delivered directly towards the customers or first needs to come towards the facility in Apeldoorn after which it gets delivered together with the backroom elements. This way of delivering depends on the country of origin of the customer.

Figure 2.2; EasyPac machine

Figure 2.3; MultiPac machine

The tables behind an MultiPac system has significant differences in comparison with tables suited behind EasyPac systems. An example of such a difference is that tables suited for the MultiPac systems have additional holes to fit the machines. Nevertheless, the table assembly becomes more challenging and takes longer for such tables. Therefore, tables for the MultiPac systems take remarkably longer to produce, especially when these tables also have an L-shape.

2.1.2 Roller belt production

Roller belts form an essential part of the backroom of the bottle deposit systems. Roller belts can guide empty drinking packages and handed-in crates towards further parts of the backroom for collection. TOMRA assembles these roller belts by hand within a separate production department.

The roller belts consist of one or multiple so-called 'mats'. Each mat consists of eight rolls, which are put into a frame to complete a single mat. The roller belts are being produced in lengths of two meter, one meter, half a meter and a quarter of a meter. These lengths require four mats, two mats, one and one half a mat for their production respectively. Figure 2.4 shows an example of a roller belt consisting of several mats.

Figure 2.4; Roller belt

Additionally, the empty drinking packages sometimes need to be guided towards a different direction within the backroom of the bottle deposit systems for optimal space allocation. TOMRA makes corners of roller belt for these occasions, allowing the empty drinking packages to take turns within the backroom. Corners of 90 degrees, 45 degrees and 30 degrees are being produced to realize the turns. Figure 2.5 provides a visualisation of a 90 degree corner.

Figure 2.5; 90 degree corner

At the roller belt department, a variety of other standardized product are produced as well that can help with guiding empty drinking packages towards further places for sorting and collecting. Most of these products consist of automated roller belts, capable of transporting empty drinking packages in a more controller manner or towards different elevation levels. Table 2.1 provides an overview of the items produced at the roller belt production department, including a brief description as well as the variants per product type.

Table 2.1; Roller belt production overview

Product name	Description	Variants
Roller belt	Transport empty drinking packages and crates on so-called 'mats' by means of gravity	▪ 2m ▪ 1.5m ▪ 1m ▪ 0.5m
Corner	Change direction of roller belt within the backroom	▪ 30 degrees ▪ 45 degrees ▪ 90 degrees
German Corner	Similar to regular corner, but contains an adaptation to the design that only applies to corners from German orders	▪ 30 degrees ▪ 45 degrees ▪ 90 degrees
Elevator belt	Lift items on roller belts to transport the items further via gravity	▪ 2m ▪ 1.5m ▪ 1m ▪ 0.5m
Elevator belt start/stop	Similar to a regular elevator belt, but includes a built-in start/stop system	▪ 2m ▪ 1.5m ▪ 1m ▪ 0.5m
PET elevator belt	Guides sorted empty PET bottles towards a collecting location or bag	▪ 160mm width ▪ 470mm width
Merger	Guide handed-in crates from a deposit machine towards the roller belts	▪ Single size 1.75m
V2 Conveyor	Situated immediately after the bottle deposit system and is capable of transporting empty drinking packages	▪ Single size 1 m
Eco-Wall	Deposit system for a diversity of recyclable materials such as batteries or lamps	▪ T8/T9 600 ▪ T9 E+L+B ▪ T9 300

All products that require production at the roller belt department are being produced within standardized formats, as previously described. This allows the roller belt parts to be produced all the time so that they are being made to stock. In other words, the roller belts can be produced before an order arrives (if time allows this) and are stored within the finished goods warehouse. The exact number and types of roller belt needed to complete an order is available. So, to fulfil an order, the required roller belts can be picked within the finished goods warehouse to deliver these parts together with the other backroom elements. It can also occur that orders contain of only parts from the roller belt department.

2.1.3 Conveyors production

An incoming order often requires the production of one or multiple conveyors. Conveyors are able to guide the empty drinking packages in a more controlled manner compared to the roller belts. Conveyors consist of an automated belt, which is able to transport the empty drinking packages towards different heights. TOMRA assembles these conveyors by hand from a set of raw materials. The conveyors are of different lengths and heights, which are specified roughly beforehand by their customers. Especially the length is of great influence on the production time. Because of these differences within the conveyors, TOMRA produces the conveyors only when an order arrives. In total, TOMRA produces five products at the conveyor department, four conveyor types and one additional product that enhances user-friendliness during disposing empty drinking packages.

Table 2.2 provides a brief description per produced product at the conveyor department.

Table 2.2; Overview of produced elements at the conveyors department

Product name	Description	Figure
Z-shape	Can elevate empty drinking packages towards a different height in a Z-shape (see Figure 2.6)	 *Figure 2.6; Z-Shape conveyor example*
Feeder	Transports large quantities of drinking packages towards a different height (see Figure 2.7)	 *Figure 2.7; Feeder conveyor example*
Pre-feeder	Allows for transportation of large quantities of empty drinking packages towards further stages (see Figure 2.8). As the name implies, often situated before a feeder conveyor	 *Figure 2.8; Pre-feeder conveyor example*
Depot	Is able to guide empty drinking packages towards a different height, where there is no flat part located at the top	N.A.
Wash & Waste	Washing sink and disposal bin in the shape of bottle deposit device to enhance user friendliness (see Figure 2.9)	 *Figure 2.9; Wash and Waste*

Customers often order one or multiple conveyors often together elements from the table and roller belt department to create a complete backroom system. However, sometimes orders occur consisting of solely products that require production at the conveyor department.

2.2 Planning and scheduling activities

This section outlines how TOMRA manages its incoming orders to generate a production plan of the production departments. First, we provide an overview of the IT system that TOMRA uses to supports its production processes in Section 2.2.1. In addition, we discuss the current way TOMRA plans and schedules the incoming orders. A distinction is made between two different scheduling levels, namely planning on tactical level and on operational level which are described in Sections 2.2.2 and 2.2.3 respectively.

2.2.1 IT system

TOMRA uses IFS as the ERP system for their production processes. IFS contains information regarding all the incoming orders. For each order, the system knows the different backroom elements required (table, roller belt and/or conveyor). The production specifications are derived from the bill-of-materials of the order element. This way, the system directly knows which materials are required to complete every element of an order. IFS also couples the assigned due dates of an order to the different order elements, so that an overview can be created on which backroom elements require soon production.

IFS currently lists the production activities that can be worked on per department at a specific moment in time, where the activity with the earliest due date is listed at the top. TOMRA puts a lot of trust in this system and only sees the production steps of an order when the due date approaches, i.e., when the production activities are located at the top of the list. It is noticeable that currently, the order due date is coupled to all elements of that specific order. Consequently, production on an order takes place when the due date approaches which results into problems during peak demand periods.

Furthermore, IFS acts as an inventory control system, as the system allows TOMRA to see how many items are in stock for each raw material as well as for the finished goods. The items on stock for each product are changed when a part of an order is transferred or set to complete. The system knows the materials required for the completed order part or element, so that the inventory levels are updated. Operational managers and some employees can indicate complete orders elements within the system.

Additionally, the system automatically indicates when materials needed for production activities reach a certain reorder point. When this happens, TOMRA orders the required material(s) from their suppliers, where the number of items ordered is based on an order-up-to level. However, the operational managers at TOMRA also monitor the inventory for materials themselves. By looking currently approximately 120 days ahead in time, which is flexible and can change over time, the managers assess whether to order new materials. They do this on daily basis by observing the current inventory levels, orders within the system and outstanding orders that are not delivered yet.

Figure 2.10 provides a simplified overview of the way in which IFS supports the production activities at TOMRA.

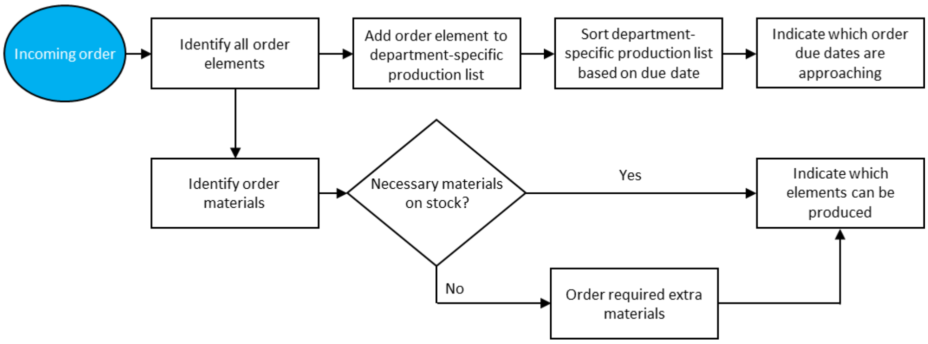

Figure 2.10; Overview IT system support

However, IFS does not provide some sort of feedback on the feasibility of the assigned order due dates. This implies that the system does not consider external delivery times from the suppliers of raw materials for assessing the due date feasibility. This can result in materials not being available when the due date is approaching or when production is scheduled. Especially for stochastic external lead times, this often can cause inventory shortages and not meeting due dates.

2.2.2 Tactical level

The way a tactical level production plan is generated by TOMRA is that order due dates determine the outline of the schedule. When orders arrive, customers have the most saying into the order due date during the order negotiation, which are often stated a couple of months ahead in time. TOMRA does have some control in this process if they feel that this will not fit within their current production capacity or with respect to their suppliers, however this control is minor. To maintain a strong competitive position within the market and to satisfy the needs of customers, TOMRA wants to hold on to the proposed due dates as much as possible. However, TOMRA does want to have more saying into when a due date is feasible or not by means of capacity related overviews.

When a due date is assigned for a certain order, the tactical level plan is more or less complete. In order words, the tactical level plan is completely based on the due dates of the incoming orders. These assigned due dates give an outline on when which order will be produced. As mentioned before, the ERP system lists the required production activities per department based on the order due date. Since TOMRA focuses a lot on this ERP system, orders are being produced close to their due date instead of spreading (part of) the workload towards earlier periods.

As the ERP system does not consider required and available capacity, problems are likely to arise. For instance, one can imagine that the production workloads suddenly can increase significantly when relatively large orders have due dates close to each other or when there are many order due dates within a short time span.

Figure 2.11 visualizes an overview of the workload spread for the table production of the year 2022 from available data. This figure shows the total number of tables produced per week. Neglecting the production during the first and last weeks of the year, large fluctuations in the workload per week occur. As TOMRA produces only produces the tables when due date approach and assigns these due dates in a non-optimal manner, the workload is not spread evenly throughout the weeks.

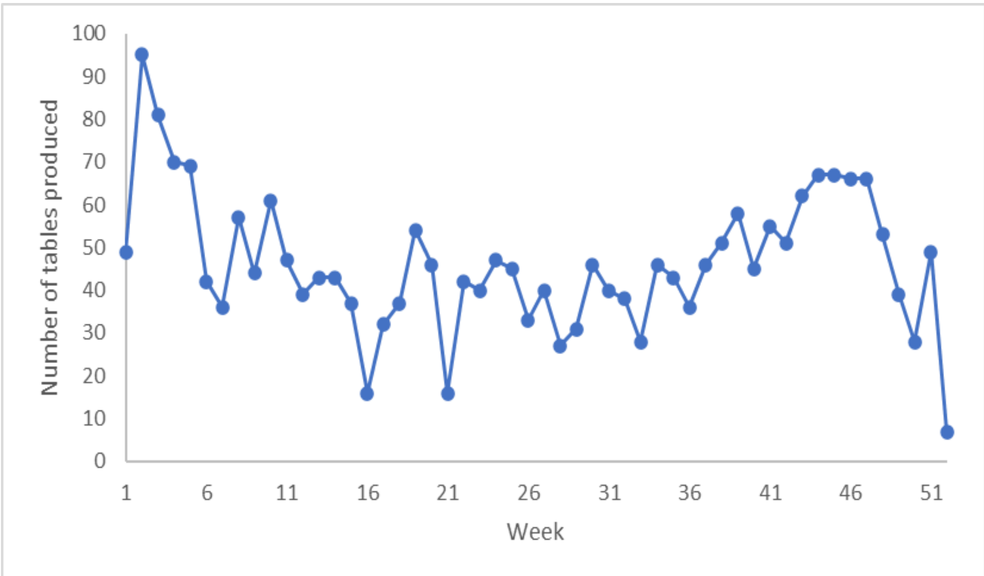

Figure 2.11; Workload spread table production 2022

In addition, as a consequence of waiting to produce the backroom elements until just before the assigned due date and having no capacity related insights, TOMRA can never provide earlier due dates to customer orders then the ones that their clients request. On the other side, TOMRA does not refuse customer orders to maintain a strong market position, customer satisfaction and good relationships. This all can result in a significant rise in the workloads in a short time period. Employees at the workplace often work in overtime to be able to finish all orders on time during these workload intensive periods. The costs of working in overtime, however, do not transfer over into the prices charged for customer orders. This indicates that costs of working in overtime are for TOMRA themselves.

There are, however, two important reasons why some production activities only take place close to their due date. The first of which being that TOMRA has limited space within their finished-goods warehouse, where finishing production activities well ahead of the due dates would result in storage related problems. Especially for the tailor-made products that TOMRA produces at the table and conveyor departments, it is not desired to finish production early.

Secondly, at TOMRA, there is a clear distinction between Dutch orders and foreign orders. The TOMRA facility in Apeldoorn receives and handles order requests from Dutch clients. When such a request arrives, TOMRA has limited information regarding the exact specifications of the order. For the requested standardizes elements of the roller belt department, TOMRA knows the number and specifications of the corresponding roller belt elements within an order at the moment of its arrival. However, when it comes to tables and conveyors that Dutch customers request, TOMRA only knows the number of tables or conveyors included within an order and has a general idea of the types requested. TOMRA knows the exact specifications regarding sizes and shapes of tables and conveyors within Dutch orders only about one week before the customer-requested due date. This is because all Dutch customers finalize the table and conveyor designs together with TOMRA after several checks on the designs at the location of the backroom installation.

The TOMRA facility in Apeldoorn does not directly receive order requests from foreign customers, as these arrive at foreign TOMRA locations. These foreign TOMRA departments stay in close contact with foreign customers and finalize the order specifications together with their customers. After this has been done, the foreign TOMRA department requests a foreign order to the production facility in Apeldoorn, where all exact order specifications are known and production can commence immediately. However, due to the first reason, these foreign orders are often produced close to their assigned due date.

The two reasons for the production of orders close to their due dates suggest the importance for optimizing due date assignments for incoming orders. Namely for a well-coordinated production and prevention of production capacity related issues.

2.2.3 Operational level

Approximately one week before the individual due dates of the tables and conveyors of a single order, an operational production schedule is constructed by the operations managers. So, production of an order only takes place a week before the stated individual product deadlines. During peak demand periods, production starts especially late as a consequence of the busy schedule and the scheduling per week approach. If a large order occurs, then part of the order is scheduled a couple of days earlier to spread a part of the workload. The operational managers know that they should start earlier on these large orders, as they have a rough idea what types and sizes of order are approaching their due dates and feel like they might run into trouble if they wait with the production of these large orders.

Additionally, the production of tables and sometimes conveyors for Dutch clients can take place at most a couple of days before the customer requested due dates, which is different from other (foreign) customers. As mentioned before, the reason for this is that the backroom design for Dutch clients need to be checked first at location, which often takes place at a very late stage. The operational production schedule is made by hand with logical thinking as the way of scheduling.

The operational schedule allows for products of different orders to be produced non-sequentially. For instance, products for different orders can be produced in parallel on multiple workstations. Furthermore, the operational managers try to schedule the production of workload heavy systems in the beginning of each week. Again, no scientific reasoning lies behind this, only that employees will have most energy to work on the largest tables and conveyors at the beginning of each week.

An overview of the way of planning at TOMRA on a tactical level an operational level can be seen in Table 2.3.

Table 2.3; Overview of scheduling levels

Level	By whom	Task	Time horizon
Tactical	Customers	Assigning order due dates	Monthly
Operational	Operational managers	Determining job sequences	Weekly

2.3 Conclusion on current situation

The production process of the backroom elements of TOMRA consists of three main production departments: table production, roller belt production and conveyor production. Tables and conveyors can only be produced once an order arrives due to customizability of these elements. In addition, parts from the roller belt production department are standardized and can be produced upfront.

From the current way of operating, it becomes clear that there is a need for proper due date assignment of incoming orders as well as a need to spread workloads. TOMRA is very dependent on their ERP system, which lists the necessary productions based on the earliest due date. However, this ERP system does not consider required and available capacity. This results in capacity related issues during peak demand periods, since TOMRA waits until the very last moment to produce asked elements. This implies that, next to optimizing order due date assignments, the workloads should be spread more evenly on a tactical level.

3. Literature review

Within this chapter, we provide a literature review related to the combined due date generation and workload spread problem. This chapter forms the theoretical framework of the research. Section 3.1 provides an overview of different types of production environments and elaborates on the required activities in the context of TOMRA. Section 3.2 addresses due date assignment methods from the literature. Section 3.3 discusses relevant tactical level resource allocation methods. In Section 3.4, we elaborate on ways to cope with production uncertainties in the context of this research. Section 3.5 presents an overview of heuristics to tackle optimization problems. At last, Section 3.6 provides a summary of the literature review.

3.1 Production environments

Various types of production environments follow from the literature. Lager (2003) makes a general distinction between four different types of manufacturing environments, namely make-to-stock (MTS), assemble-to-order (ATO), make-to-order (MTO) and engineer-to-order (ETO). MTS concerns with standardized products in which production planning can be based on forecasted demand. On the other hand, ETO engineers and produces customizable products from scratch which differ per customer. MTO consists of producing customized products, where some or all of the necessary production to complete an order takes place after receiving the order (Saniuk & Waszkowski, 2016). ATO assembles the customer orders from standardized products and are seen as a combination of MTS and MTO production environments.

The production environment of TOMRA can be seen as make-to-order (MTO) manufacturing, as the production of tables and conveyors only take place after a customer places an order. Since TOMRA is situated within MTO manufacturing, this type of environment is applicable for this research.

Figure 3.1 provides a hierarchical framework for planning activities in such a MTO environment. This framework depicts planning activities related to multiple time horizons. The strategic level deals with activities in the long term (e.g. one or multiple years ahead), whilst the tactical level considers decisions in the mid-term (e.g. multiple weeks or months ahead) and the operational level deals with decisions in the short-term (e.g. on a daily or weekly basis).

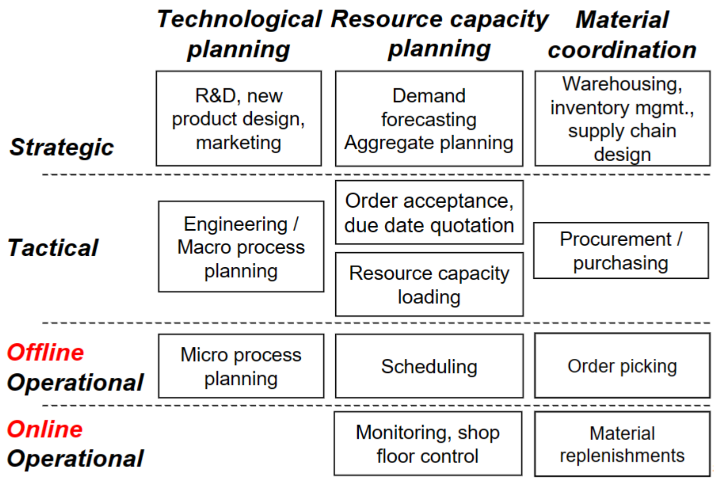

Figure 3.1; Hierarchical planning framework applied to MTO (Hans et al., 2007)

Decisions such as the quotation of due dates and assigning available production capacities towards orders (resource capacity loading) are located on under the resource capacity planning header at the tactical level on the framework. This implies that resource capacity planning should serve as input to assign due dates and spread the workloads of incoming orders on the resources appropriately. In order words, it is especially relevant to consider the production capacity in making such decisions.

3.2 Due date assignment methods

This section discusses due date assignment methods found within the literature. First, Section 3.1.1 treats a general view on how due date assignment problems are solved. Next, Section 3.1.2 talks about due date assignment models from the literature that are focused on production environments similar to TOMRA. At last, Section 3.1.3 discusses several relevant finite loading methods that are capable of assigning due dates.

3.2.1 General view on due date assignment

The topic of due date assignment, also referred to as due date quotation, is well-discussed within the literature. Assigning advantageous due dates for customer orders and timely delivery of goods to customers is an important aspect in production environments and is said to enhance customer satisfaction and provide a competitive advantage (Sha & Liu, 2004). More specifically, in order to avoid tardiness penalties, with the possibility of losing customers, production companies are under increasing pressure to assign attainable due dates (Jing, 2013). On the other hand, promising due dates too far in the future may not be acceptable by customers, forcing a company to offer price discounts to retain business or resulting in lost sales (Jing, 2013). Due date assignment can thus be seen as challenge where a trade-off occurs between assigning short due dates to incoming orders and preventing tardiness penalties for not meeting due dates.

There exist plenty model variants for due date assignment, where assigning due dates is considered a necessary step for scheduling problems in production environments. After due dates are assigned, a production schedule is often constructed to meet these established due dates, indicating that assigning due dates often goes hand in hand with creating a production schedule (Ng et al., 2003).

Gordon et al. (2002) introduce several relevant due date assigning methods. The most found relevant models are addressed. First of all, Gordon et al. (2002) discusses due date assignment based on the total work content (TWK) of an incoming production order. This is a convenient and relatively simple model, where larger production orders are assigned a due date further in time. The TWK due date assignment method assigns due date d_i order i based on the release date r_i. This release date indicates the first possible start time of an order, i.e., when all required materials of an order are available for production. A multiple of the order processing time p_i is added to the release date in the TWK method in the following manner:

$$d_i = r_i + kp_i$$

The constant $k > 0$ represents a common multiple. This constant has to be determined in order to define the due dates for each order. The job release date r_i causes jobs only to be produced once its desired raw materials are available to start the production.

In addition, Gordon et al. (2002) mention due date assignment based on the number of operations (NOP) required to complete a job. Within this model, jobs requiring more operations to be completed are assigned a due date further in time. The NOP due date assignment method assigns due date d_i to order i based on a multiple of a constant model parameter $k > 0$ and the number of operations required M_i to complete order i as follows:

$$d_i = r_i + kM_i$$

At last, Gordon et al. (2002) discusses a due date assignment method based on common slack (SLK). Here, due dates are assigned according to a sum of the release dates r_i, processing times p_i and a common slack value q in the following way:

$$d_i = r_i + p_i + q$$

The common slack factor stimulates consistency when, for instance, all jobs have to be delivered to the same customer.

The methods discussed by Gordon et al. (2002) are relatively simple and indicate that the assignment of a due date depends on the workload of an order as well as when an order or job can start its production. Moreover, these models are especially useful when due date assignment decisions are necessary in the context of scheduling problems, where order sequencing can be addressed based on the assigned due dates. However, in the specific production environment of TOMRA, these models cannot directly be considered useful.

3.2.2 Due date assignment in MTO

Not many due date assignment methods in the context of MTO manufacturing are available within the literature due to production uncertainties and complex production systems. According to Zorzini et al. (2008), the higher the product complexity and customization degree, the more workload-based methods and detailed analyses are needed to assign due dates.

First, due dates in MTO production environments can be assigned based on the bottleneck operation. Production processes in MTO manufacturing can be dominated by a bottleneck operation stage (Ten Kate, 1994). This bottleneck operation can be used to assigned due dates, since the throughput of an entire production facility can depend on the bottleneck process. Park et al., (1999) propose a Heuristic Delivery Due Date Algorithm (HDDDA), allowing decision makers to find reliable due dates for each production order in a MTO environment by considering the residual capacity of the bottleneck operation. The heuristic introduced by Park et al. (1999) assigns a target production date to an order, after which a production starting date is determined via backwards scheduling of the order. If this starting date precedes the order release date, then the target production date is changed, otherwise the starting date is feasible. Next, the workload of the bottleneck process is compared with the maximum capacity of this bottleneck process. If the workload of the bottleneck process is less than its maximum capacity, then the due date is easily determined, otherwise the heuristic changes the target production date and starts again.

Due dates in a MTO manufacturing environment can, moreover, be assigned based on the total production capacity. Production capacity can be considered for due date assignment decisions by looking at resource availabilities, previously accepted orders as well as their production progression (Guhlich et al., 2015). This implies that a detailed capacity overview is necessary for these decisions. Moreover, Guhlich et al. (2015) state that decisions on due dates should occur directly after an order has arrived.

Corti et al. (2006) propose a capacity-driven model to verify the feasibility of proposed due dates of customers and establish reliable due dates in a MTO environment. The method compares the requested capacity of an incoming order with the actual available capacity. A proposed due date can either be stated feasible or infeasible by the model based on two capacity checks. In detail, once a due date is proposed, the earliest release date (ERD) and last operation completion date (LOCD) are known. The ERD is obtained by adding the maximum delivery lead time estimated of the needed materials to the arrival date of the order request. While the LOCD equals the due date minus the expected time for quality control and order finalizing steps. Two schedules are constructed via forward and backward scheduling for an incoming order. When applying forward scheduling, the available capacity is allocated starting from the earliest release date and first operation based on average production times (Zorzini et al., 2008). Whilst backwards planning allocates the available capacity starting from the set due date and last operation of the production cycle based on average production times (Zorzini et al., 2008). These methods create the following lower and upper bounds for each operation j of order i:

- The earliest starting date (ESD_{ij});
- The earliest completion date (ECD_{ij});
- The last starting date (LSD_{ij});
- The last completion date (LCD_{ij});

These bounds are used to calculate the capacity that is requested on each production process in a relevant time horizon. This is done by adding the overlap between different orders to indicate if an order due date is feasible based on a critical index. This critical index indicates whether a resource will be overloaded, i.e., the maximum capacity is likely to be exceeded. Per time horizon, which could be one or multiple weeks, the possible maximum and minimum requested capacity on a certain resource s as well as the available capacity on this resource are determined to form the critical index.

In addition, MTO production environments cope with various uncertainties during the order acceptance stage according to Wullink et al. (2004). In this paper, uncertainty with respect to used materials, total work content, tool requirements, production times and resource availabilities are considered. As the order information is available in detail for TOMRA during the order acceptance, there are no uncertainties with regard to the materials required for an order. Variabilities lie within the production times as well as for external lead times of raw material. Song et al. (2002) develop a method of assigning due dates taking into account uncertainties in production times. This method assigns a due date for an order when the probability of completing the order before the due date is equal to a certain service target level λ.

$$F_w(d_i) = \lambda,$$

where $F_w(d_i)$ indicates the cumulative probability distribution of the completion time of order i with due date d_i. This cumulative probability distribution is the convolution of the truncated cumulative distributions of the production stages (operations to complete an order). Distributions such as exponential or normal are assumed for the processing times of every operation j required for completing order i. These convolutions, however, depend on many underlying assumptions making the results highly dependent on the choice of distribution.

3.2.3 Finite loading methods

Finite loading models combine due date setting with capacity planning and consider workloads of incoming orders for this process. Finite loading methods assume a fixed capacity available and do not permit overloads. Various finite loading models existing are capable of assigning due dates in a MTO environment.

The way finite loading methods work is that, first, the planning horizon is broken down into T equal length time buckets with capacity norm or maximum regular capacity Q_{st} for resource s in period t. An incoming order i arrives with its corresponding operations j, which need to be loaded into the time buckets. The operations of an order are added (loaded) to the existing workload WL_{st} of resource s in period t. Each operation j of an order i has a processing time p_{ij}, so when an operation is loaded into a time bucket, its workload increases by this processing time. Finite loading methods have as output due dates d_{ij} that indicate the due date of operation j of order i. The due date of the entire order is equal to the due date of the last loaded operation of that order.

Thürer et al. (2013) discuss a finite loading approach that considers the concept of forward scheduling in assigning due dates, namely Forward Finite Loading (FFL). Note that there is a difference between scheduling and loading, since loading assigns work to a set of resources but does not directly indicate in detail when and in which order operations are carried out. The FFL method determines due dates in a stepwise manner by fitting required operations of an order to remaining capacity. Figure 3.2 visualizes an example of loading workloads within a periodic loading framework. Starting with the first operation of a job, this operation is loaded on resource s if there is enough capacity to load the full workload of this operation in period t. Otherwise formulated, if $WL_{st} + p_{ij} \leq Q_{st}$, then the operation if fully loaded into that period. If there is not enough capacity available in the corresponding period, then the next period $(t + 1)$ is considered until the operation can be loaded fully into a single period. Note that d_{ij} is the end of the time period in which operation j of order i is loaded. However, this does not necessarily have to be the case, as different choices can be made for setting the due date. For instance, one can include some form of slack, where d_{ij} is some time after the time bucket of the last loaded operation instead. At last, d_{i0} indicates the release date time of an order.

If there are precedence relationships applicable, i.e., another operation or production step needs to take place beforehand, then time periods after the due date of that previous operation are considered for loading the operation. More specifically, an operation can only be loaded from d_{ij-1} onwards if precedence relations apply, where d_{ij-1} indicates the due date of the preceding operation. The due date of such an operation is then the end of the first time period in which the operation with processing time p_{ij} can be loaded.

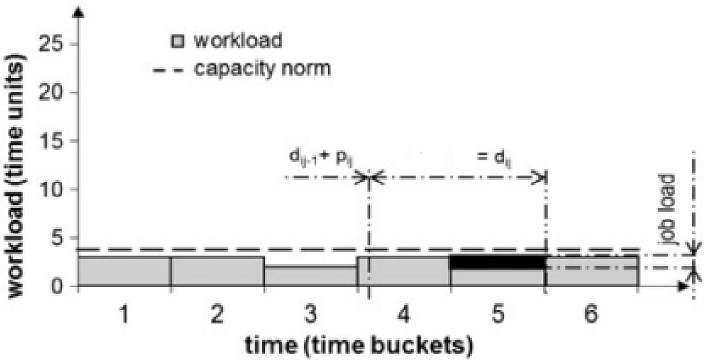

Figure 3.2; Finite Forward Loading (Thürer et al., 2013)

The FFL model has several limitations. First, the operations cannot be partially loaded into multiple time buckets. This implies that a processing time can never exceed the capacity norm. In addition, the due date of a job is set to the last loaded time bucket. Since it is unknown which loaded operation will be produced first, the due dates might be overestimated.

To overcome the partial loading limitation (i.e. operations must be fully loading into a single time bucket) of the FFL approach, Robinson and Moses (2006) introduce a search procedure that allows for partially loaded operations. For an operation j of order i , the number of time bucket intervals required (b_{ij}) can simply be determined as follows:

$$b_{ij} = \left\lceil \frac{p_{ij}}{G} \right\rceil$$

where p_{ij} is the corresponding processing time and G indicates the time bucket size (granularity). b_{ij} is rounded up to assess the required number of time bucket intervals. If the number of time bucket intervals required equals one $(b_{ij} = 1)$, then the search procedure locates the earliest available single interval to insert the task. If the number of time bucket intervals is more than one $(b_{ij} > 1)$, then the search procedure looks for availability across multiple contiguous intervals.

The first advantage of the method from Robinson and Moses (2006) is that by limiting the number of time buckets in which operations can be loaded, it prevents the operations of the same job from begin spread out in a large number of time periods. In addition, the method schedules the operations in consecutive intervals if more than one interval is required.

There exist ways to deviate from the capacity norm limitation of the FFL, as stated by Nyhuis and Filho (2002). They state that forward scheduling can determine the due date from the current date, however capacity must be adjusted if this due date violates the desired customer due date. Thürer et al. (2013) discuss a Forward Finite Loading method that considers schedule deviations (FFLSD). Here, schedule deviations, rather called backlogs, are defined as the workload that should have been completed minus the actual complete load. This backlog is distributed over the time buckets. If the backlog is positive, then the backlog hours fill the available capacity gaps starting from the first bucket (earliest job release date). On the other hand, if the backlog is negative, the workload loaded in the time buckets is reduced and the available capacity is increased.

Next to the discussed FFL method, Thürer and Stevenson (2019) discuss a Backward Finite Loading method (BFL). This method uses the concept of backwards scheduling in order to operation due dates to incoming orders. When applying BPL, a due date for an order is already available, after which loading takes place backwards from this already quoted due date to determine when every operation can take place. The goal of BPL is to assign operation starting dates r_{ij} for operation j of order i.

Starting with the last operation of a job, this operation is loaded on resource s during period t if there is enough capacity to load the full workload of this operation in this period ($WL_{st} + p_{ij} \leq Q_{st}$). If there is not enough capacity available in period t, then a next period is considered until fully loaded. The release time r_{ij} on resource s is equal to the start of the period where the first operation is loaded. If precedence relationships are applicable, then the release time of the next operation (r_{ij+1}) should occur after the release time of the previous operation plus its processing time ($r_{ij} + p_{ij}$).

Nyhuis and Filho (2002) mention that backwards scheduling calculates the order release date by means of backwards scheduling from the customer due date. However, if this calculated release date lies in the past or is an unacceptable date for the customer, then capacity needs to be adjusted.

Bertrand (1983) discusses another finite loading method, namely the Cumulative Forwards Finite Loading (CFFL) method. This approach is similar to the FFL method, however a cumulative load is applicable. This implies that cumulative capacity within a time bucket at a resource is used to load the workloads of order. This cumulative load allows for operations to be spread out over multiple time buckets, where the load of each operation contributes to the cumulative load until a capacity norm is reached.

To load operation j of an order i and a certain resource s , we search for period t in which all following periods can fully load this operation ($WL_{st^*} + p_{ij} \leq Q_{st^*}$, where $t^* = t, t+1, \dots, T$). Once this period t is found, the required operation is loaded into all subsequent period t^*. The due date of operation j of order i (d_{ij}) is equal to the start of period t plus an arbitrarily selected 0.25 times the length of the period. Once again, d_{i0} indicates the release date time of an order and if precedence relationships applicable between jobs, then only time periods after the due date of that previous operation are considered.

Variants of the CFFL method exist that are capable of deviating from the existing capacity. Thürer et al. (2013) mention a Cumulative Forward Finite Loading method that considers schedule deviations (CFFLSD). Just like the previously discussed FFLSD method, the CFFLSD approach uses the idea of backlogs to deviate from the capacity norms. Figure 3.3 visualizes this approach within a cumulative loading framework.

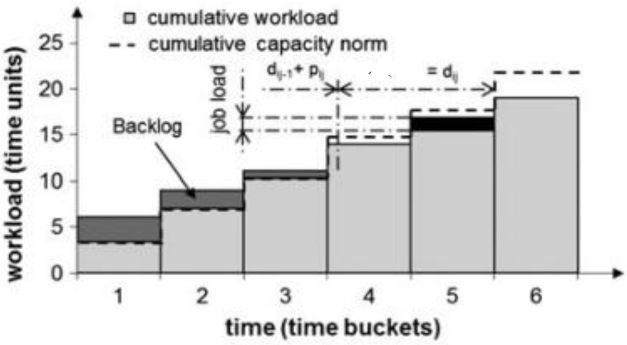

Figure 3.3; Cumulative Finite Forward Loading (Thürer et al., 2013)

3.3 Tactical level capacity planning

This section outlines knowledge from the literature regarding resource allocation to spread workloads taking into consideration the production capacity. Section 3.2.1 discusses (near)optimal capacity planning methods. Section 3.2.2 addresses commonly used objective functions for the capacity planning methods.

The assignment of due dates is often directly related with the construction of a production schedule. Due dates are often assigned during the order acceptance stage, after which is tactical level planning is made, where a tactical planning is concerned with allocating available resources to arriving orders as efficiently as possible (Wullink et al., 2004).

3.3.1 Capacity planning methods

Several methods exist within the literature that can spread the workload of incoming orders efficiently by assigning the capacity of resources. Decisions regarding capacity assignment and the use of regular and non-regular capacity are usually made at the rough-cut capacity planning (RCCP) level (de Boer, 1998).

De Boer (1998) proposes a RCCP approach for multiple projects with given order characteristics (such as due dates). Here, a set of N work packages or jobs ($j = 1, \dots, N$) that should be planned on S available resources ($s = 1, \dots, S$). Here a work package or job relates to an operation as part of an order. The time horizon is divided into T time buckets of one week, where T is large enough to potentially process all work packages. Each resource s has Q_{st} hours capacity available in every time bucket. Moreover, each work package j requires q_{js} capacity on resource s for its processing. At most a fraction $\frac{1}{\hat{p}_j}$ of work package j can be performed in any time bucket, where $\hat{p}_j$ indicates the minimum duration of work package j. Besides, work package j has release date r_j and due date d_j, so work package j cannot start before r_j and should be finished before $d_j + 1$. In addition, precedence relationships between work packages can be included, where a set of direct predecessors P_j must be completed before work package j can begin.

A proportion x_{jt} of package j in time bucket t is performed and should be decided, where an equal fraction is spent on work package j in week t on all resources. An adaptation of this exists, as mentioned by Gademann and Schutten (2005), where a fraction x_{jst} of package j on resource s in time bucket t should be decided (not necessary an equal fraction spent on all resources).

Two main variants of the RCCP problem can be identified from de Boer (1998):
1) The resource driven RCCP problem, where no nonregular capacity use of resources is allowed and the objective is to minimize the maximum job lateness. Figure 3.4 provides an example of a resource driven RCCP problem solution, where jobs with durations are all scheduled within the boundaries of the maximum capacity.

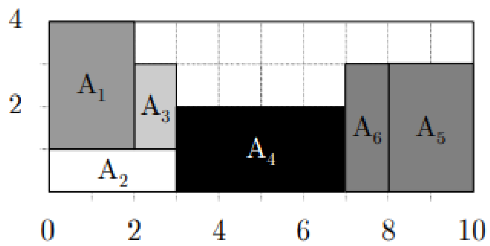

Figure 3.4; Resource driven RCCP example from de Boer (1998)

2) The time-driven RCCP problem, where nonregular capacity may be used to meet the order deadlines and the objective is to minimize the usage of nonregular capacity. Figure 3.5 gives an example of a time driven RCCP solution configuration, in which some jobs are scheduled outside the boundaries of the maximum capacity in order to meet due dates.

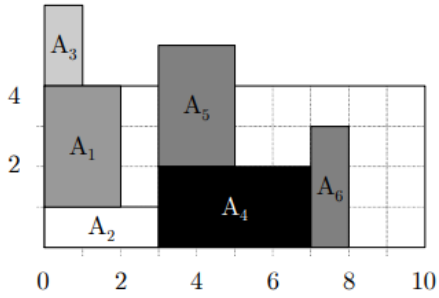

Figure 3.5; Time driven RCCP example from de Boer (1998)

There exist multiple heuristics that can find solutions to the RCCP problem variants. De Boer (1998) provides an incremental capacity planning algorithm (ICPA) for the time driven RCCP problem. Besides Gademann and Schutten (2005) propose several heuristics for the RCCP problem. They categorize the heuristics in constructive heuristics, heuristics that convert to a feasible solution and heuristics that can improve on feasible solutions.

Next to the RCCP problem, Hans (2001) mentions a resource loading problem capable of optimal resource allocation. Within this approach, both regular and nonregular capacity can be allocated, making it a combination between the resource driven approach and time driven approach discussed previously. The aim of the resource loading problem is to minimize the costs of nonregular capacity usage (working in overtime of hiring temporary staff) and the costs of lateness of jobs.

The proposed resource loading problem by Hans (2001) is stated to be NP hard. Therefore, Hans (2001) proposes a branch-and-price technique that can solve the mixed integer linear programming (ILP) model of the resource problem. This branch-and-price technique is a combination between branch-and-and column generation to find feasible solutions for the problem.

3.3.2 Capacity planning objectives

The capacity planning methods discussed above can efficiently spread the workload on resources on a tactical level according to various objective functions. We discuss these objective functions in more detail within this subsection. Besides, we discuss other relevant objectives as identified from the literature.

The resource driven RCCP problem strives to allocate workload to resources in such a way that the lateness of a job is minimized, where a job relates to an operation of an order. Job lateness L_j is defined as $C_j - d_j$, with job completion time C_j and due date d_j. In addition, job tardiness T_j can be written as $max\{0, C_j - d_j\}$. These two measures can form multiple objective functions as mentioned by de Boer (1998) and Hans (2001):

- Average tardiness : $\frac{1}{N}\sum_{j=1}^{N} T_j$
- Maximum tardiness : $\max_{j=1,\dots,N} T_j$
- Number of tardy jobs : $\sum_{j=1}^{N} v(L_j)$, where $v(x) = \begin{cases} 1, if\ x > 0 \\ 0, otherwise \end{cases}$
- Average lateness : $\frac{1}{N}\sum_{j=1}^{N} L_j$

- Maximum lateness : $\max_{j=1,\ldots,N} L_j$

For the resource driven RCCP problem formulation, the objective is to minimize the maximum lateness of all related jobs.

The time driven RCCP problem aims to minimize the costs of the use of nonregular capacity. Gademann and Schutten (2005) formulate the objective function of the time driven RCCP in the following way:

$$min \sum_{t=1}^{T} \sum_{s=1}^{S} c_{st} U_{st}$$

where there are T time buckets, S resources and one unit of nonregular capacity usage U_{st} on resource s during week t has a cost of c_{st}. The goal is to minimize the costs of nonregular capacity usage $c_{st}U_{st}$ on all resources over all time buckets. However, de Boer (1998) argues that organizations also want to avoid peaks in the capacity usage. Therefore, the paper suggests to squaring the nonregular capacity usage on resource s during week t instead (U_{st}^2).

In addition to minimizing lateness, tardiness, the costs of nonregular capacity usage or a combination between them as proposed by Hans (2001). Gordon et al. (2002) suggest that one should also include costs for completing an order or job too early. In this paper, they define job earliness as $max\{0, d_j - C_j\}$, where d_j indicates the due date and C_j the completion time of job j.

3.4 Production uncertainties

This section discusses production uncertainties that commonly take place in a MTO production environments. Section 3.4.1 addresses variabilities in the lead time of suppliers. Moreover, Section 3.4.2 elaborates on ways to deal with processing times variability in a scheduling context.

3.4.1 Lead time variability

Variability might occur within external lead times of suppliers from needed raw materials. External lead time variability results in uncertain release dates of orders (Wullink et al., 2004), which can cause production related issues such as not meeting due dates. Every reasonable effort should be made in order to eliminate lead time variability, however often external lead time variability relates to uncertain shipping times or inconsistent suppliers (Silver et al., 2021).

Silver et al. (2021) mention the application of external lead time distributions to cope with the external lead time variability. For this, estimates of the mean $E(L)$ and variance $Var(L)$ of the external lead time length are necessary to determine. These parameters can be determined via data analysis, where often a Normal distribution follows. However, according to Silver et al. (2021), when the ratio of standard deviation of the external lead time length and expected external lead time demand $\frac{\sqrt{var(L)}}{E(L)}$ becomes large, then a Gamma distribution is more appropriate. The main reason for this is that otherwise negative values of the lead time length might occur.

However, if no clear external lead time distribution is distinguishable, then sampling method by means of Monte Carlo simulation are applicable to cope with lead time variability. Shapiro (2003) mentions the sample average approximation (SAA) method that is capable of solving stochastic optimization problems. The idea behind this approach is that scenarios are randomly sampled, where the objective function consists of the expected value over all scenarios (Shapiro, 2003). The objective function converges to the true (optimal) value if the number of scenarios approach infinity, as this is not feasible, the method uses a smaller number of scenarios to approximate the value. In addition, Pagnoncelli et al. (2009) discuss a way to solve stochastic optimization problems using SAA where the uncertainty lies within the constraints instead. This is applicable when it is desired to acquire a feasible solution, where different scenarios can test the feasibility.

3.4.2 Processing time variability

Variability of processing times is a common characteristic of MTO production environments. Here, the challenge is to estimate the processing times of (aspect of) incoming orders as precise as possible to avoid capacity related issues.

A way to cope with variable or uncertain processing times is to estimate them by means of a probability distribution. Shen and Zhu (2018) assume production processing times to follow a normal distribution for a parallel machine scheduling problem. Moreover, Goren and Sabuncuoglu (2009) assume processing times to be distributed according to a uniform distribution in the context of a single machine scheduling problem.

Machine learning techniques can also be applied to deal with processing time variabilities in a scheduling context. Techniques such as using artificial neural networks or regression are especially relevant when it comes to this (Yamashiro & Nonaka, 2021). These machine learning techniques can estimate processing times from a set of input parameters. Input parameters can relate to the type of products produced, order sizes or steps required. In addition to machine learning techniques, data mining can also serve to estimate order processing times from the order attributes (Öztürk et al., 2006).

3.5 Heuristics

Finding exact and optimal solutions for capacity planning problems in manageable computational times is not feasible for large problem instances (de Boer, 1998; Gademann & Schutten, 2005). In order to find good or near-optimal solutions in reasonable computational times, approximate methods, also called heuristics, are often of use (Aickelin & Clark, 2011). The literature highlights two main categories of heuristics, namely construction heuristics and improvement heuristics. We discuss promising construction and improvement heuristics for our problem in Sections 3.5.1 and 3.5.2 respectively. At last, Section 3.5.3 highlights relevant neighborhood operators for the problem at hand.

3.5.1 Construction heuristics

Construction heuristics can make complete solutions from scratch in an efficient manner. This type of heuristic starts with an empty solution and iteratively adds solution elements to extend the solution until a complete feasible solution is reached (Sörensen et al., 2018). Often, a priority rule or otherwise called greedy selection rule determines which element(s) to add in each iteration (Sörensen et al., 2018). In a scheduling context, elements can for instance refer to jobs or operations that need to be assigned to a set of resources or machines. These jobs can iteratively be scheduled on the available resources within a construction heuristic to form a complete solution. This can be done based on a wide range of priority rules, such as First Come First Served (FCFS), Earliest Due Date (EDD), Shortest Processing Time (SPT) etcetera (Ruiz, 2015).

Some types of construction heuristics, such as dynamic programming, are even capable of generating optimal solutions by recursively constructing solutions. In addition, construction heuristics allow no backtracking. This implies that an element cannot be removed from a solution once this part of the solution is selected (Radar, 2010).

3.5.2 Improvement heuristics

Improvement heuristics aim to find a better solution from a feasible starting solution in an iterative manner. This is done by constructing neighborhood solutions, where neighborhood solutions can be constructed by modifying the current solution to a certain extent. Section 3.5.3 addresses several relevant ways in which a neighborhood solution can be constructed. Several improvement heuristics exists that can cope with neighborhood solutions in various ways, where eventually the best found solutions is returned.

The steepest descent heuristic searches within the neighborhood $N(x)$ of a certain feasible solution x to find solution x' with the best objective value. If such a solution x' exists, then this best neighbor solution is selected and the process is repeated. The main advantage of the steepest descent heuristic is that is it capable of finding better solutions relatively fast (Pirlot, 1996). There are some drawbacks to this steepest descent heuristic, since first, the best found solution is likely to be a local optimum instead of the desired global optimum as depicted in Figure 3.6. This is because the quality of the best found solution greatly depends on factors such as the initial solution, the way a neighborhood solution is chosen or the neighborhood for each feasible solution (Pirlot, 1996; Rader, 2010). Moreover, the steepest descent heuristic stops once a local optimum is found, so the heuristic stops if there is no improving solution within the neighborhood.

There are heuristics that can avoid being stuck at local optima. These so-called metaheuristics can balance intensification and diversification to overcome the local optima (Radar, 2010). Intensification refers to exploiting areas with promising solution characteristics, whereas diversification allows exploration of a wider area of possible worse neighborhood solutions. Steepest descent focuses on intensification, however metaheuristics can balance between the two.

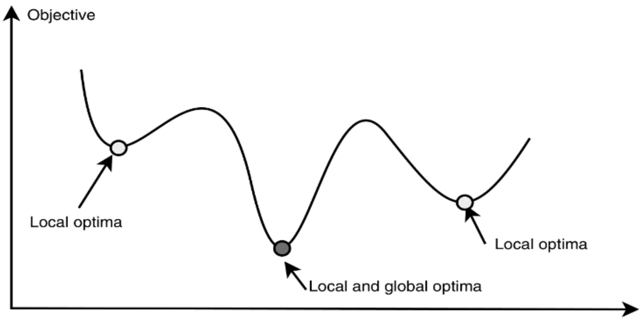

Figure 3.6; Local and global optima visualization (Tahasozgen, 2021)

Simulated annealing (SA) is such a metaheuristic capable of escaping local optima by allowing hill-climbing moves (Henderson et al., 2003). The general idea of SA is that the algorithm starts with diversification and ends with intensification. The way SA works is that neighborhood solutions are constantly constructed for a prespecified number of iterations (Markov chain length). A constructed neighborhood solution x' is accepted when its objective is better than the current solution x. If such a solution is accepted, then the heuristic searches in the neighborhood of the accepted solution for other solutions during the next iteration. However, when a neighborhood solution x' is worse, the solution is accepted with a certain probability. This acceptance probability depends on the difference in objective value between the two solutions ($f(x) - f(x')$) as well as the progression of the heuristic represented by temperature T. The temperature level decreases after the Markov chain length, which often happens by means of cooling factor α (where T becomes αT). This lower temperature results in lower acceptance probabilities of neighborhood solutions. In other words, the process of decreasing the temperature results in more intensification and less diversification as the heuristic progresses. After decreasing the temperature, the algorithm starts again with the construction of neighborhood solutions for the duration of the Markov chain length, but now with the lower temperature. In addition, the overall best found solution is stored throughout running the algorithm and updated if a better solution is found.

The heuristic stops when a certain condition is met. This condition can be a certain temperature level, a prespecified number of iterations or when the current solution does not change after a prespecified number of iterations (Radar, 2020).

Figure 3.7 provides an overview on how the SA heuristic works.

```
Generate initial solution x
Set x* = x
While stopping condition not met:
    For m = 0 to MarkovLength
        x' = random solution in N(x)
        If f(x') is better than f(x):
            x = x'
            If f(x') is better than f(x*)
                x* = x'
            End
        Elseif random number(0,1) ≤ e^((f(x)−f(x'))/T)
            x = x'
        End
    End
    Update T
End
Result = x*
```

Figure 3.7; Simulated annealing heuristic

The choice for the parameters of the SA heuristic, namely the Markov chain length, the initial temperature level, the way of updating the temperature and the stopping condition is referred to as the cooling scheme.

Tabu search (TS) is another type of metaheuristic, where memory is used to escape from local optima (Glover & Laguna, 1993). TS applies the sample principle as local search, where the best neighborhood solution is selected until a local optimum is reached. TS can move away from these local optima, as recent solutions are put on a tabu list and cannot be selected until they are removed from this list. This limits the neighborhood of the solution and prevents cycling behavior to recently visited solutions (Radar, 2010). Already visited solutions can be removed from the tabu list if the list exceeds a specified maximum length, indicating how long solutions are kept in memory. In summary, the best solution that is not on the tabu list is selected from the neighborhood of the current solution.

Solution characteristics are commonly stored on the tabu list, since storing entire solutions can be impractical due to large solution sizes (Radar, 2010). Many solutions can have the stored solutions characteristics on the tabu list, including the global optimum. To prevent making better solution tabu, an aspiration criterion can be implemented that allows tabu moves to be used. Such a criteria could be that a tabu move may be used if it results in a better solution than the current best found solution. Note that TS stores the overall best found solution as well.

The heuristic stops when a certain stopping criterion is met, which can be a prespecified number of iterations or a maximum running time (Glover & Laguna, 1993).

Figure 3.8 shows an outline on the way the TS heuristic operates.

```
Generate initial solution x
Set x* = x
Tabu list = Ø
While stopping condition not met:
    Generate neighbor solutions from N(x)
    For x̂ in neighbor solutions
        If x̂ ∉ Tabu list and f(x̂) is better than f(x)
                x = x̂
        Elseif Aspiration criterion met
                x = x̂
        End
    End
    If f(x) is better than f(x*)
        x* = x
    End
    Add x to Tabu list
    If Tabu list length > Max Tabu list length
        Remove last from Tabu list
    End
End
Result = x*
```

Figure 3.8; Tabu search heuristic

3.5.3 Neighborhood operators

The choice of using an appropriate neighborhood operator or neighborhood search technique that is capable of generating neighborhood solutions is crucial for the performance of improvement heuristics (Glover, 2003). One or multiple neighborhood operators define the neighborhood structure, which is the set of solutions that can be reached from current solution x in one single step of the algorithm (Glover, 2003). In addition, a neighborhood is connected when the used neighborhood operator(s) can convert any solution configuration into another solution, including a global optimum, within a finite number of iterations.

Table 3.1 shows the found relevant neighborhood operators that are identified from the papers of Glover (2003), Guo et al. (2014) and Krim et al. (2019) in the context of scheduling jobs or operations of orders onto a set of resources or machines.

Table 3.1; Overview neighborhood operators in scheduling context

Neighborhood operator	**Definition**
Swap operator	The neighborhood structure of the swap operators consists of all solutions that can be obtained by swapping jobs with each other within a schedule. This implies that the swap operator maintains the positions within the schedule, where only the type of job changes. This operator allows that all jobs can be swapped with each other to generate a new neighborhood solution.
Consecutive swap operator	The consecutive swap operator obtains neighborhood solutions by only allowing swaps between consecutive scheduled jobs.
Pairwise swap operator	This type of operator generates neighborhood solutions in a similar way as the swap operator, but swaps a pair of consecutive scheduled jobs rather than a single job.
Move operator	The neighborhood structure of this type of operator consists of all solutions that can be obtained by removing a job and moving the job into another position within the current schedule. This operator allows a job to be moved in between two other jobs.
Couple move operator	This type of operator is similar to the move operator, however, moves a pair of consecutive scheduled jobs to another position within the schedule to obtain neighborhood solutions.
Inverse sequence operator	The neighborhood structure of the inverse sequence operators is made out of all solutions that can be obtained by selecting two jobs and inversing the sequence in between.

Note that some actions of the neighborhood operators might result into infeasible solutions. Therefore, verifying the feasibility of (neighborhood) solutions that are obtained via neighborhood operators is of great importance.

3.6 Literature review summary

The production environment of TOMRA can be identified as MTO. This implies that decisions such as the assignment of due dates or capacity to resources should be made whilst considering the production capacity.

The identified due date assignment methods in this chapter can help with the process of assigning due dates to incoming orders. The discussed methods from Gordon et al. (2002) are relatively simple and cannot be used directly, since these methods do not consider production capacities in setting due dates. However, from these methods, it becomes clear that the order due date is highly dependent on the workloads as well as the time at which the production can begin.

We identified due date assignment methods applicable in a MTO environment. First, the algorithm proposed by Park et al. (1999) assigns due dates based on the bottleneck operation, but it does not consider required and available capacity on all resources. Moreover, the method from Corti et al. (2006) consists of some relevant bounds to calculate the requested capacity for every production process to check the feasibility of already quoted order due dates. This method, however, is not able to generate order due dates. Furthermore, Song et al. (2002) use cumulative probability distributions of processing times to deal with production uncertainties for assigning due dates. As the process of establishing these probability distributions as well as the results depend on many underlying assumptions, these are not desirable for usage. Nevertheless, the proposed idea of using a certain service target level to deal with uncertainties within the production is considered useful.

Especially the finite loading methods are deemed relevant, since these methods consider available and occupied capacity for assigning due dates. From the identified methods, the FFL model and shows solid performance on the lead time and tardiness of incoming orders after a simulation study conducted by Thürer and Stevenson (2019). Another reason for preferring the FFL method is that this method can generate a clear overview of the loaded operations per week. In addition, Robinson and Moses (2006) describe a way to spread out the loaded operations over multiple weeks. This approach is not directly useful, but the idea that operations can be loaded into multiple consecutive intervals can make the FFL method more convenient to use.

The finite loading methods, however, are not capable of optimally planning the required workloads on the resources. That is why we address methods that can efficiently construct a tactical level capacity plan. The methods discussed from de Boer (1998), Hans (2001) and Gademann & Schutten (2005) are able to construct such a plan for a set of orders with (pre)defined characteristics. However, as the customers of TOMRA arrive one by one, due dates should be assigned in the same manner. Within the literature, an approach that is capable of assigning an individual order due date whilst optimally (re)allocating capacity on a set of resources is lacking. Nevertheless, the discussed approaches mention some relevant aspects, where nonregular capacity can be used to meet order due dates. Additionally, we address several objective functions for the construction a tactical level capacity plan. The literature makes a distinction between lateness, tardiness and the use of nonregular capacity, which are all considered helpful.

Production uncertainties play a big role in MTO environments. We address literature related to the variability in external lead times from suppliers of raw materials as well as variability in production times. When it comes to the uncertain production times, uncertainties take place at the table production and conveyor production in terms of sizes and shapes. Machine learning techniques mentioned by Yamashiro and Nonaka (2021), such as regression are especially relevant for determining production times (or workloads) for these products from a set of order characteristics. Silver et al. (2021) address ways to assign distributions to the external lead times from supplies of raw materials, which is a way to incorporate uncertainty. Moreover, the SAA method from Shapiro (2003) and

Pagnoncelli et al. (2009) are especially relevant for solving stochastic optimization problems when no clear distributions are appliable. However, the incorporation of external lead time variability of necessary raw materials into due date assignment and/or spreading the workload of an incoming order is not available within the literature.

As all previously identified capacity planning problems are unlikely to find an optimal solution in reasonable computational time, we identify a set of heuristics from the literature. Construction heuristics can construct feasible solutions from an empty solution by means of a greedy selection rule (Sörensen et al., 2018). These heuristics are relevant to efficiently construct a feasible solution from an empty solution. In addition, we address three types of improvement heuristics from the literature, namely steepest descent, simulated annealing and tabu search. The latter two can escape from local optima and are, therefore, more likely to result into a better solution (Henderson et al., 2003; Glover & Laguna, 1993). The overall best approach for a type of heuristic and the way to construct neighborhood solutions for the problem setting at TOMRA is hard to identify from the literature.

4. Modeling approach

This chapter describes the modeling approach of the research. Section 4.1 presents an outline of the purpose of the model and discusses decisions that we make to transform the problem at hand into a model. Section 4.2 highlights the modeling assumptions applicable. In Section 4.3 we present the deterministic model formulation including model constraints and the objective. Section 4.4 discusses a way to incorporate stochastic external lead times within the model formulation. At last, Section 4.5 provides a conclusion to the chapter.

4.1 Model outline

This section provides an outline of the model that we construct for solving the combined due date assignment and workload spreading problem at TOMRA. We mention the purpose of the model together with a global outline on how to realize this. Besides, we globally discuss the thoughts behind the model approach as well as decisions that we make to model the problem at hand.

The main purpose of the model is to optimize due date assignment to incoming orders of customers. The model should make these tactical-level decisions based on the required production capacity of an incoming order as well as on already occupied production capacity of existing orders that have not been completed yet. To realize this, it might be beneficial to impose lateness to the due dates of existing orders or make use of nonregular production capacity, i.e., working in overtime or temporary hiring extra staff to increase the capacity. This implies that, as a consequence, our approach regulates the spread of the workloads on a set of resources.

The finite loading methods from the literature discussed in Chapter 3 can help us in realizing the purpose. These methods are the most applicable when it comes to assigning due dates to incoming orders whilst considering required and occupied production capacities (Thürer et al., 2013). The way finite loading methods work is that workloads of incoming orders are loaded within certain time periods. This loading occupies capacity within such a period and the maximum capacity cannot be exceeded. These methods look in which periods the workloads of incoming orders can be loaded, where the last loaded period determines its due date.

However, we require some adaptions to the finite loading methods from the literature for the problem at hand. For instance, these methods only allow the workloads of entire orders to be loaded within a single certain time period and does so if there is enough capacity available. This implies that the finite loading methods do not consider an optimal spread of the workloads. There is a variant that allows loading the workload within multiple time periods when this workload exceeds the maximum capacity (Robinson & Moses, 2006). Nevertheless, the finite loading methods still do not generate an optimal spread of the existing and incoming workloads, as the purpose of the method is solely to generate due dates and not the optimal loading of resources.

Methods that can spread the workload of a set of orders efficiently are capacity planning methods related to RCCP problems that we discussed in Chapter 3. These methods consider the use of nonregular capacity to meet deadlines (time-driven) as well as imposing lateness to the due dates of existing orders (resource-driven) for spreading the workloads. The overall objective for these RCCP problems is to minimize the use of nonregular capacity, the costs of lateness of orders or a combination of the two (Hans, 2001). The principles of both the resource-driven and time-driven RCCP problems can be used during due date assignment of incoming orders.

4.1.1 Workload representation

The modeling approach that we discuss within this research assigns due dates to incoming orders and does so by loading the incoming workload of an order to a set of resources in an efficient manner. This workload loading is done in such a manner that the approach also considers reloading the existing workloads of already arrived orders to create loading opportunities for the incoming order. As mentioned in Chapter 3, there is a clear difference between loading and scheduling. This is because loading assigns workloads of individual orders to a set of resources but does not directly indicate in detail when and in which order these workloads are carried out. Eventually, the outcome should provide an outline of the loaded workloads of individual orders over various time periods. This can provide insights on a tactical level related to available capacities, extending due dates and the usage of overtime.

Within this subsection, we introduce the way in which we represent workloads that we use during our modeling approach. While doing so, we use visualizations to bring across ideas behind the model. In further stages of our research, we continue to use visualizations with the same workload representation that we introduce here.

Customers make order requests, which have a certain workload that can be translated into a production duration, from which the amount depends on the order request. Moreover, when a customer requests an order, there are already orders, each with their own workloads, within the system that have not been completed yet. Orders consist of one or multiple operations, which we define as the different requested elements that together account for the entire order request. In the setting of TOMRA, such an operation could be a tailor-made table or conveyor. This implies that we disregard the different steps it takes to complete the operations themselves, i.e., all steps it takes to complete an operation together belong to the same single operation. The workload of every individual operation within an order together form the total workload of such an order. Figure 4.1 depicts an example of the workload of an order. The order within this figure consists of four operations, each with their own individual workload, that together account for the total workload of the order.

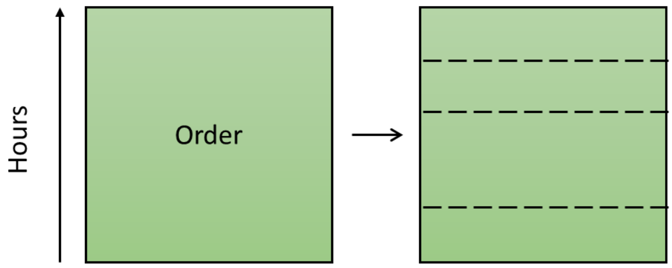

Figure 4.1; Workload of an order

Orders can consist of multiple operations, each of which requires production at a distinct resource. We refer to a production department instead of resource for the remaining of this research, where each department has their own production capacity. Since an operation only requires production at one department in our setting, we separate the total workload of an order into the workload per production department. The workload of an order at a certain production department, thus, depends on the operations that require production at this department. We state the workloads of orders or operations within orders as hours in our approach, as, in our view, this provides sufficient detail to load workloads related to production capacities on a tactical level.

For representing and loading the workloads of orders at a production department, we can make use of the loading frameworks from the literature. These loading frameworks represent the available production capacity as well as the capacity that is occupied by workloads of orders during a time period at a certain department. Within these frameworks, time periods or time buckets make up for the horizontal axis and the workload represents the vertical axis (see Figure 4.2).

We identified two loading frameworks within the literature, namely a periodic framework and a cumulative framework. The periodic framework, mentioned by Thürer et al. (2013), represents the production capacity per time period at a department. On the other hand, the cumulative framework represents the cumulative capacity per time period at a department (Bertrand, 1983). Figure 4.2 provides an example of the periodic framework (left within the figure) and the cumulative framework (right within the figure). This example visualizes the workloads of five loaded orders, each with a different coloring, and their corresponding operations by means of the workloads representation from Figure 4.1. Every order is loaded into a single time period for simplicity, where a department has a certain maximum capacity during every time period.

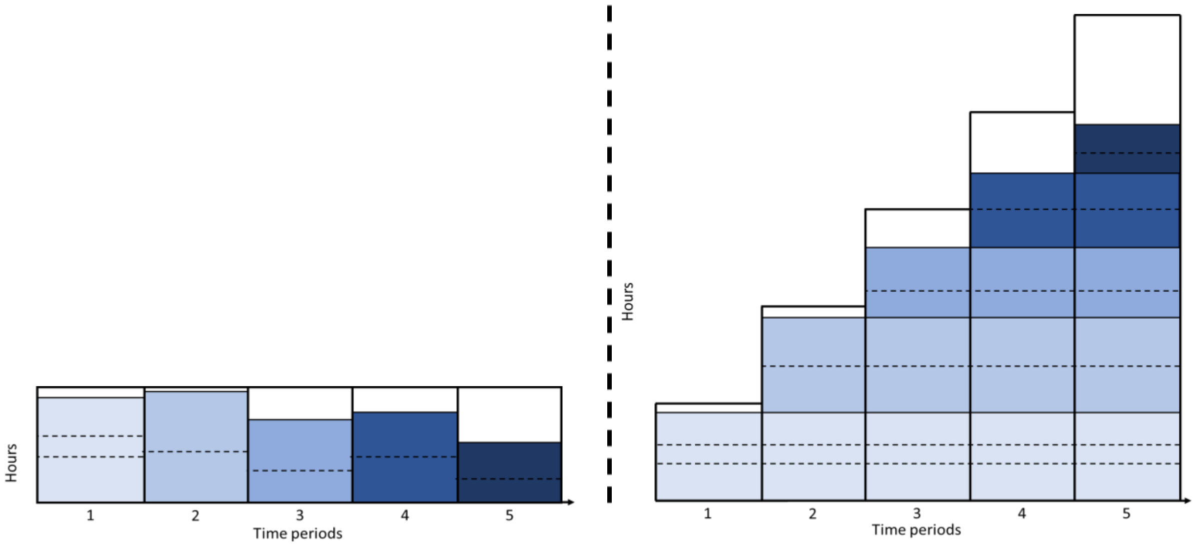

Figure 4.2; Loading frameworks periodic (left) and cumulative (right)

Within the literature, both these frameworks are applied to finite loading methods where the entire workload of an order must be loaded into a single time period (Bertrand, 1983; Thürer et al., 2013; Thürer & Stevenson, 2019). Within this setting, the cumulative framework has as advantage that it accounts for more interaction between successive periodic planned workloads. This implies that within the cumulative framework, workloads are not exactly linked to a certain period and can be carried out in other periods as well. Where the order loaded into period 1 in the periodic loading framework in Figure 4.2 should be carried out in this first period, the cumulative framework allows this order to be carried out in later periods as well. This advantage can result in more loading opportunities, as period 5 has more capacity available in the cumulative framework within Figure 4.2 compared to the periodic framework.

However, the cumulative framework representation can easily become cluttered when there are many orders loaded, whereas the periodic framework gives a clear overview of the loaded orders per period. Moreover, the advantage of the cumulative framework is mainly due to the specific setting within the literature, where the workload of an order must be loaded into a single period. The only variant for this is that it can be allowed to spread the workload of orders over multiple periods if the workload of a single order exceeds total capacity of a period (Robinson & Moses, 2006). Within our setting, we also include the possibility of loading operations of an order in different periods. We elaborate on the exact ways in which we load the workloads of orders in Chapter 5. Besides, TOMRA desires to have a clear overview of which exact orders or operations of order occupy the capacity within a certain period. Because of these reasons, we choose to use the periodic framework for loading workloads of orders in further stages of this research.

4.1.2 Loading decisions

The purpose of the model is that it should help in deciding the best way of when to load which workloads of orders per department. This relates to loading the incoming order as well as possible reloading of already existing orders. Figure 4.3 provides a simplified example on choices that a solution approach for solving the model should consider. This figure illustrates two solution instances, where the workloads of multiple orders are loaded within a periodic loading framework containing multiple time periods t with a regular capacity and maximum overtime hours (which are the same for all periods here). These time periods are set to weeks with a time horizon dependent on the loaded workloads in the future. A reason for this is that we identified, from the literature in Section 3.1, that decisions such as assigning due dates and resource capacity loading should occur at the tactical level (multiple weeks or months ahead). A choice could also have been to use time periods of days instead, however, this provides too much detail, especially when loading further ahead in time. Moreover, we consider weeks instead of days as TOMRA desires to be flexible with scheduling operations of orders on a daily basis and wants the possibility to decide this themselves.

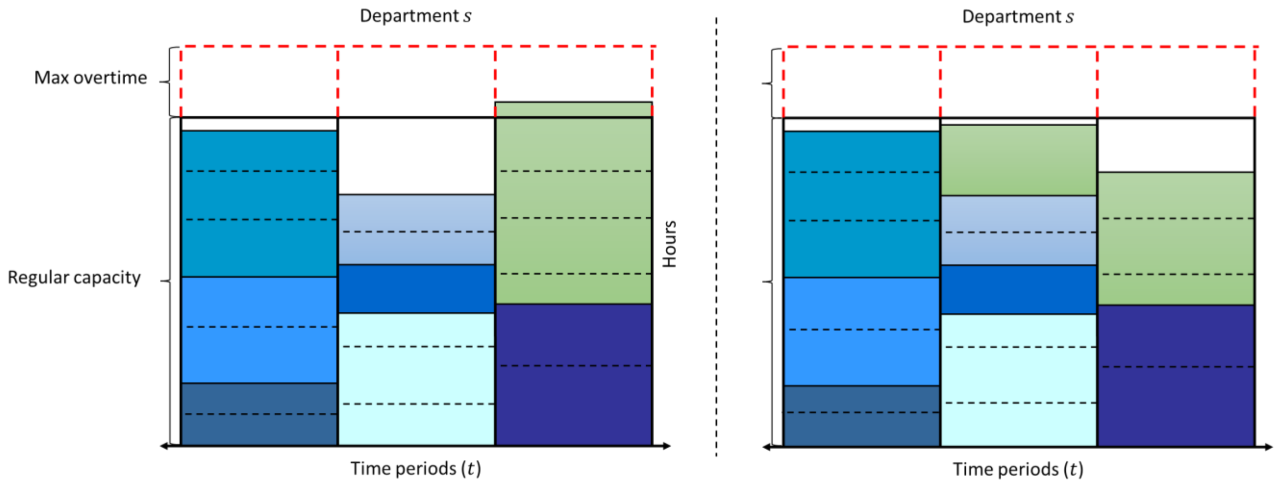

Figure 4.3; Model loading choices

Within the figure, each color box represents the workload of a different order with its individual operations that is loaded within a certain period at this department s. As explained before, the size (height) of each order relates to its workload or production time. The different shades of blue boxes in the figure indicate already loaded orders at a certain production department over multiple time periods. The green box relates to the workload of the incoming order.

The solution instance on the left in Figure 4.3 loads the entire workload of the incoming order in a single period. However, it is also a possibility to spread the workload of the incoming order over multiple time periods. The instance on the right in Figure 4.3 depicts this principle, where we split the workload of the incoming order and load it in two different time periods.

We identify two options when it comes to splitting the workloads of orders, either we split the total workload of an order based on its operations or allow splitting the operations themselves. Figure 4.4 visualizes these two options.

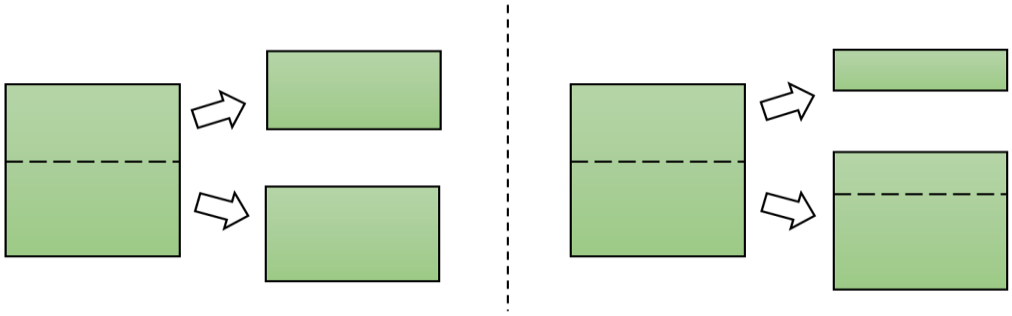

Figure 4.4; Splitting workloads by operations (left) and splitting operations (right)

The previously discussed example from Figure 4.3 uses the approach where we split the workload of an incoming order at its exact operations (left situation in Figure 4.4). The main advantage of this approach is that this way, operations are coupled to specific periods instead of possibly spreading individual operations over multiple periods. The advantage of splitting the operations themselves is that this way, we can regulate the capacity usage by splitting the workloads more accurately. However, allowing splitting the operations themselves could mean that we should investigate the earliest moments when parts of the operations can commence its production instead of the entire operation. Because of these reasons, we decide to split the workload of an order based on its operations for our approach.

The previously discussed example from Figure 4.3 does not consider that other orders can be reloaded as well, which can be very useful for creating opportunities to load the incoming order in other periods. Figure 4.5 provides a simplified example for the same setting as before, however this time, an already loaded order is reloaded as well. The example illustrates that the incoming order can now be loaded in an earlier period. Because of this, we can possibly provide an earlier due date to the incoming order, however we increase the workloads by means of using nonregular capacity in two of the three visualized periods to realize this. The model should provide us insights into the best way of loading the orders, where the model considers costs or penalties for several aspects such as using nonregular capacity or assigning an earlier or later due date.

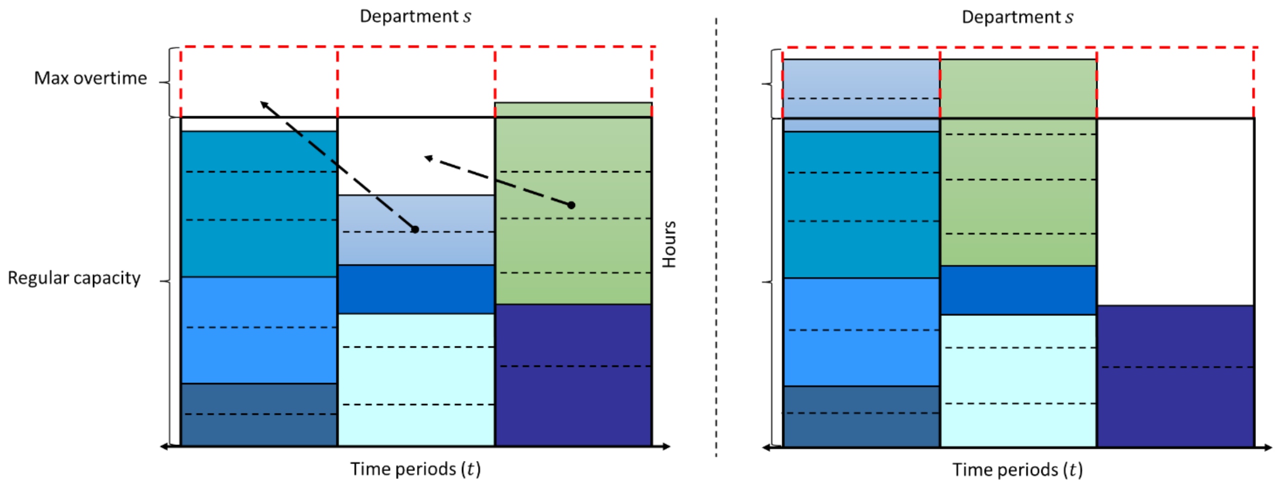

Figure 4.5; Model reloading choice

4.2 Model assumptions and simplifications

This section mentions the assumptions and simplifications that we make during the modeling approach. These assumptions and simplifications influence the way we solve the problem at the research company.

We consider the following model assumptions for the arrival of customer orders:

- Customers make order requests one at a time, where each customer proposes a customer-requested due date for their order. We moreover assume that the incoming orders arrive independently without patterns, unrelated to the demand of other customers.
- Customer orders are never rejected when they arrive. This implies that when customers arrive at TOMRA with order requests, due dates should always be assigned to the incoming customer orders.

For the way of operating to complete orders, we make the following assumptions:

- The production capacity can only be increased by working in overtime. We do not consider nonregular capacity by means of temporary hiring staff or outsourcing production activities.
- We make no distinctions between employees that are responsible for the production at TOMRA. Each employee on the work floor works on a fulltime basis to the standards of TOMRA during a week. This also implies that every employee has the same maximum numbers of overtime hours in a week.
- The employees at the production departments complete operations from orders without interruptions or failures. In other words, we do not consider production breakdowns at the work floor for the processing times.
- Operations should be loaded into one single week.
- The workload of individual operations never exceed the combined regular and nonregular capacity of any week.
- Operations require production at one single department.
- There are not precedence relationships between different operations within orders.

We consider the assumptions below for the costs that we incorporate within the model:

- If customer-requested due dates or already assigned due dates are not met, we charge costs, where we regard all customers the same and make no distinction between different customers. This implies that the costs included for tardiness of orders are the same for every customer.
- Costs included for working in overtime are fixed and the same for every department and for every time period.
- The earliness costs related to assigning a due date that precedes the customer-requested due date can differ per customer. This is because customers might already have coordinated storage or transport planned based on their request when they arrive, meaning that they do not desire an earlier due date. On the other hand, it might occur that customers see benefits in an earlier due date. We assume that the preferences of customers related the earliness are known at the moment of an order arrival.
- The earliness costs related to finishing all the production activities of an order before the already assigned due date of that order can differ per production department. For some departments, it might be desired to finish the production early to account for small order requests at that department for other customers. However, earliness can also be undesired for a certain department as this can result in large storages of finished products.

4.3 Model formulation

Within this section, we formulate the deterministic model regarding the problem at hand. We address the model sets, parameters and variables. Moreover, we state and elaborate in detail on the model constraints and objective function.

4.3.1 Problem setting

During the arrival of a new customer order request, we have a set N consisting of orders (index i), each with their own operations j from the set J_i to be loaded on a set of departments S (index s). Each operation j requires production at department y_j (where $y_j \in S$). We refer to the hours it takes to finish an operation as its processing time p_j, which make up the workload of an individual operation. We divide the time horizon in time periods of one week, where the set T (index t) contains the future time periods.

We make a division between the incoming order, which we denote by i', and the existing orders that have not been completed yet ($i \neq i'$). The incoming order has a customer-requested due date in period $cd_{i'}$ and each existing order i has an already assigned due date in period ad_i.

Right before the new incoming order request, each operation j in order i was loaded in period z_j. The periods where these operations are loaded can change to fit the workload of the incoming order efficiently. Let x_j denote the period where operation j is loaded after the arrival of a new order request. x_j cannot precede its release date r_j, indicating the first possible period for producing operation j. Moreover, for the operations that are part of orders that have an already assigned due date within a frozen period, which are periods within the set H (index h) indicating the first $|H|$ periods within T, we do not allow loading them in other periods than the initial z_j.

The x_j variable results in loading workloads of individual operations (p_j hours) at the corresponding departments y_j in certain time periods. We indicate to the total hours of workload at department s during period t by WL_{st}. During the process of loading operations, we consider a regular capacity of Q_{st} hours and maximum nonregular capacity of MO_{st} hours at department s during period t.

From WL_{st}, we can derive the nonregular capacity usage U_{st}. Here we impose costs CO for using overtime of U_{st} raised to the power of integer exponent Exp, where higher values of this exponent result in solutions favoring the utilization of less overtime across multiple periods rather than of having large fluctuations in the overtime hours per period.

From x_j we can derive the first and last loaded periods of a single order at each production department (b_{is} and c_{is} respectively). The last loaded operation of an order at all department can provide information regarding the due date each order i, namely d_i. Here we introduce a slack of q periods, where we assign due date of order i (d_i) q periods after $\max_s\{c_{is}\}$. Furthermore, from b_{is} and c_{is}, we can derive the time span in which operations j within each order i are loaded per production department. We impose cost SC of spreading the operations an order over multiple periods at a department.

The due date of incoming order i' ($d_{i'}$) provides feedback on the customer-requested due date $cd_{i'}$. More specifically, we can derive possible tardiness $RT_{i'}$ or earliness $RE_{i'}$ related to $cd_{i'}$, with costs RCT and $RCE_{i'}$ respectively. Additionally, the resulting due dates of the existing orders d_i can cause tardiness AT_i or earliness within a production department AE_{is} related to already assigned due dates ad_i. For the existing orders, we include cost ACT for tardiness and ACE_s for earliness.

Below, we provide an overview of the mentioned model sets, parameters and variables.

Set	Description
S	Set of production departments indexed by s, and $S = \{1, \dots, \|S\|\}$
T	Set of future time periods considered indexed by t, and $T = \{0, \dots, \|T\|\}$
H	Set of frozen time periods indexed by h, $H \subset T$, and $H = \{0, \dots, \|H\|\}$
N	Set of orders indexed by i, and $N = \{1, \dots, \|N\|\}$
J	Set of all operations
J_i	Set of operations required for order i indexed by j, and $J_i \subseteq J$

Parameter	Description
y_j	Department to produce operation j, where $y_j \in S$
p_j	Processing time of operation j
r_j	Release date of operation j
z_j	Period where operation j is loaded before arrival of new order
$cd_{i'}$	Customer-requested due date of incoming order i'
ad_i	Already assigned due date of order i
WL_{st}	Total workload at department s in period t
Q_{st}	Maximum regular capacity of department s in period t
MO_{st}	Maximum overtime hours at department s in period t
q	Slack denoted in periods for assigning due dates
Exp	Integer exponent of overtime hours
RCT	Cost of exceeding a customer-requested due date
$RCE_{i'}$	Cost of preceding the customer-requested due date of order incoming i'
CO	Cost of one hour of overtime at any department during a period
ACT	Cost of exceeding an already assigned due date
ACE_s	Cost of preceding an already assigned due date at department s
SC	Cost of loading the workload of an order over multiple periods at a department

Decision variables	Description
d_i	Due date of order i
x_j	Period where operation j in order i is loaded after arrival of new order
b_{is}	First time period in which any part of order i is loaded at department s
c_{is}	Last time period in which any part of order i is loaded at department s
$RT_{i'}$	Tardiness on customer-requested due date of incoming order i'
$RE_{i'}$	Earliness on customer-requested due date of incoming order i'
AT_i	Tardiness on already assigned due date of order i
AE_{is}	Earliness on already assigned due date of order i at department s
U_{st}	Hours of nonregular capacity usage at department s in period t

4.3.2 Model constraints

The model has several constraints, on which we elaborate on here.

$$x_j = z_j \qquad \forall i \in N | ad_i \leq |H|, \forall j \in J \tag{1}$$

Constraint (1) restricts reloading of orders within the frozen periods. These periods are the first couple of periods within the model and orders in these periods may already have been scheduled in the short term with everything arranged for the production. The set H with index h contains the time periods that fall under the frozen periods. So, we fix the already loaded orders within these periods at the moment of a new order arrival. This implies that we restrict operations of orders that have an already assigned due date within a frozen period to their original loading periods. To make sure that we do not restrict the incoming order i', its already assigned due date $(ad_{i'})$ should initially lie outside the frozen periods.

$$x_j \geq r_j \qquad \forall i \in N, \forall j \in J \tag{2}$$

Model constraint (2) guarantees that operations of orders can only be loaded in periods when the raw materials are available for production activities. We indicate the earliest moment of production of operation j within order i as its release date r_j. These release dates are equal to periods within the model and can differ per operation within an order.

$$WL_{st} \leq Q_{st} + MO_{st} \qquad \forall s \in S, \forall t \in T \tag{3}$$

$$WL_{st} = \sum_{i \in N | x_j = t} \sum_{j \in J_i | y_j = s} p_j \qquad \forall s \in S, \forall t \in T \tag{4}$$

Constraints (3) and (4) indicate that the loaded workload of orders within time periods cannot exceed the maximum regular capacity and maximum overtime of the time period at a certain department. Both the maximum capacity Q_{st} and the maximum overtime hours MO_{st} depend on the number of employees that are available during period t at department s. The loaded workload, WL_{st}, depends on the existing, already loaded, order operations in period t at department s. More specifically, WL_{st} depends on the processing times of these operations of orders. The processing times of the individual operations are an input for the model and should be known (or estimated) at the time of the order arrival.

$$b_{is} = \min_{j \in J_i | y_j = s} \{x_j\} \qquad \forall i \in N, \forall s \in S \tag{5}$$

$$c_{is} = \max_{j \in J_i | y_j = s} \{x_j\} \qquad \forall i \in N, \forall s \in S \tag{6}$$

Constraints (5) and (6) determine the first and last loaded time period of order i at department s. The model can determine these variables by means of the periods where the operations of the orders are loaded after the arrival of a new order (x_j). More specifically, the first loaded time period of an order at a department is the lowest x_j from the operations j in order i that require production at department s $(y_j = s)$. The last loaded time period of an order at a department, then, becomes the highest x_j from the operations in the order at the department.

$$d_i = \max_s\{c_{is}\} + q \qquad \forall i \in N \tag{7}$$

Constraint (7) determines the due date of all orders, so for the incoming order i' as well as for all other orders in the system. The due date of an order depends on the last period for all the production departments in which an operation of that order is loaded. Note that the due date is a time period and not an exact date within the model. We make this choice, since this way we avoid working with individual days, making the model more detailed towards an operational level instead of a tactical level. As we do not know the exact moment when operations will be carried after loading them weeks or months ahead in time, we add some slack q in assigning due dates. This slack provides some extra space within the production and avoids loading the entire capacity of a week for orders with due dates at the beginning of this week for instance.

Figure 4.6 visualizes a simplified example of solution instances of two production departments. Within this figure, orders are loaded in four periods (six through nine), where each period has its own maximum capacity and maximum overtime hours. From the loaded orders, we can determine their due dates in the way we discussed above. When looking at the workload of the incoming order i', indicated by the green boxes, we can see that these are loaded in different periods for the two departments. The periods of the last loaded order part of department 1 ($c_{i'1}$) is equal to 8, whereas for department 2, $c_{i'2}$ equals 7. This implies that the period of the last loaded order part of order i' for all departments equals 8 in this case. If we now include a slack of 1 week ($q = 1$), we assign a due date of this order in period 9. The exact date of this due date depends on the customer request.

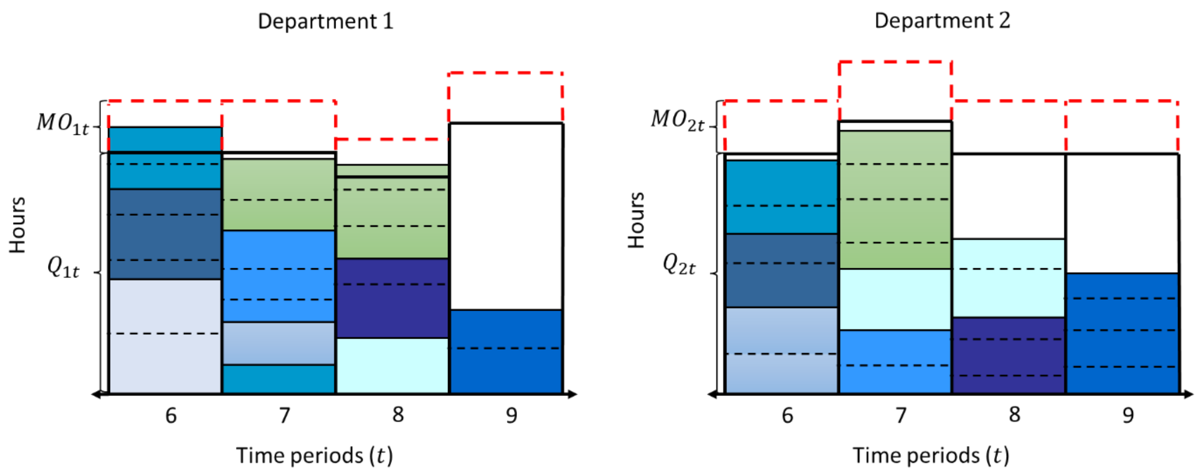

Figure 4.6; Due date assignment example

Note that we derive the latest loaded operations and with that the due dates of all other orders in the same manner. We can do this to discover if the solution causes earliness or tardiness for the orders.

$$RE_{i'} = \max\{0, cd_{i'} - d_{i'}\} \tag{8}$$

$$RT_{i'} = \max\{0, d_{i'} - cd_{i'}\} \tag{9}$$

Constraints (8) and (9) assign values to both earliness and tardiness associated with assigning a due date of the incoming order i' that deviates from the customer requested due date. Customers most likely request an exact date for their incoming order, however the model transforms this exact date to a certain time period t within the model and uses the $cd_{i'}$ parameter for this. Earliness occurs when the assigned due date of the model precedes the customer-requested due date. Tardiness takes place when the assigned due date exceeds the customer-requested due date.

$$AE_{is} = \max\{0, ad_i - (c_{is} + q)\} \qquad \forall i \in N: i \neq i', \forall s \in S \qquad (10)$$

$$AT_i = \max\{0, d_i - ad_i\} \qquad \forall i \in N: i \neq i' \qquad (11)$$

Constraint (10) defines the earliness related to an already assigned due date of order i. This earliness occurs when the period of the already assigned due date exceeds the last loaded period plus the slack of q periods for a department. The latter refers to the due date of department s, which we can compare with the already assigned due date of the entire order to see if a department finishes its production activities of an order early. Furthermore, constraint (11) measures the tardiness related to an already assigned due date. This type of tardiness takes place when the due date of the entire order i (from last loaded order fraction for all departments) exceeds the already assigned due date.

$$U_{st} = \max\{0, WL_{st} - Q_{st}\} \qquad \forall s \in S, \forall t \in T \qquad (12)$$

At last, constraint (12) measures the overtime hours during a period at a certain department. This constraint states that overtime occurs when the loaded workload in period t at department s (WL_{st}) exceeds the maximum regular capacity at that moment within the same department (Q_{st}). If this is the case, then the number of overtime hours in a period at a department is the difference between them.

4.3.3 Model objective function

The objective function consists of four parts, each indicating costs of penalties, where to aim is to minimize the total costs. More specifically, the objective function is to minimize cost related to:

1) Deviating from the customer requested due date of the incoming order i', i.e., $[RCT * RT_{i'} + RCE_{i'} * RE_{i'}]$. This deviation is associated with tardiness, assigning a due date after the customer-requested due date, or earliness, assigning a due date before the customer-requested due date. The costs related to tardiness do not differ per customer, whereas the costs linked to earliness can differ per customer as mentioned in Section 4.2.

2) Deviating from already assigned due dates of existing orders that are not completed yet, i.e., $[\sum_{i \in N: i \neq i'} ACT * AT_i + \sum_{s \in S} ACE_s * AE_{is}]$. We incur tardiness costs when a certain assigned order due date is not met, i.e. the completion time exceeds the assigned due date of an order. These costs are the same for each customer order. Next to tardiness, costs occur for being too early with the production of several elements, i.e. the assigned due date exceeds the completion time of an order. Earliness is not always desired, since this can result in large finished goods inventories and bad coordination, as mentioned in Section 4.2. Therefore, for every department we include costs related to the earliness of production activities at department s, depending on the preferences per department.

3) Spreading the workload of individual orders over a large amount of periods per production department, i.e., $[\sum_{i \in N} \sum_{s \in S} SC * (c_{is} - b_{is})]$. We include these costs, since, for instance, spreading out the workloads over a large amount of time periods of the operations within a single order that require production at the same department, where every period we load a small part of this order, is not desired.

4) The use of nonregular capacity as a result of loading the orders, i.e., $\left[\sum_{t\in T}\sum_{s\in S}\left(CO * U_{st}^{Exp}\right)\right]$. We include costs for increasing the production capacity at a certain department by means of working in overtime within a period. Note that we raise the number of overtime hours to the power of integer number Exp. Higher values of this parameter causes solutions to prefer using less overtime in multiple periods instead of having large fluctuations in the overtime hours per period.

4.3.4. Model overview

Objective function

$$\min\left\{[RCT * RT_{i'} + RCE_{i'} * RE_{i'}] + \left[\sum_{i\in N:i\neq i'} ACT * AT_i + \sum_{s\in S} ACE_s * AE_{is}\right] + \left[\sum_{i\in N}\sum_{s\in S} SC * (c_{is} - b_{is})\right] + \left[\sum_{t\in T}\sum_{s\in S}\left(CO * U_{st}^{Exp}\right)\right]\right\}$$

Constraints

$$x_j = z_j \qquad \forall i \in N|ad_i \leq |H|, \forall j \in J \qquad (1)$$

$$x_j \geq r_j \qquad \forall i \in N, \forall j \in J \qquad (2)$$

$$WL_{st} \leq Q_{st} + MO_{st} \qquad \forall s \in S, \forall t \in T \qquad (3)$$

$$WL_{st} = \sum_{i\in N|x_j=t}\ \sum_{j\in J_i|y_j=s} p_j \qquad \forall s \in S, \forall t \in T \qquad (4)$$

$$b_{is} = \min_{j\in J_i|y_j=s}\{x_j\} \qquad \forall i \in N, \forall s \in S \qquad (5)$$

$$c_{is} = \max_{j\in J_i|y_j=s}\{x_j\} \qquad \forall i \in N, \forall s \in S \qquad (6)$$

$$d_i = \max_s\{c_{is}\} + q \qquad \forall i \in N \qquad (7)$$

$$RE_{i'} = \max\{0, cd_{i'} - d_{i'}\} \qquad (8)$$

$$RT_{i'} = \max\{0, d_{i'} - cd_{i'}\} \qquad (9)$$

$$AE_{is} = \max\{0, ad_i - (c_{is} + q)\} \qquad \forall i \in N: i \neq i', \forall s \in S \qquad (10)$$

$$AT_i = \max\{0, d_i - ad_i\} \qquad \forall i \in N: i \neq i' \qquad (11)$$

$$U_{st} = \max\{0, WL_{st} - Q_{st}\} \qquad \forall s \in S, \forall t \in T \qquad (12)$$

4.4 Stochastic external lead times

TOMRA deals with stochastic external lead times for a selection of their raw material suppliers, which can have a significant impact on the production processes. For example, late deliveries of materials can directly result in not meeting already assigned order due dates. Consequently, this can lead to customer dissatisfaction and high costs. Within this section, we discuss an approach to cope with stochastic external lead times in our problem setting. This implies that we include stochasticity and expand the deterministic model from Section 4.3.

Stochastic external lead times of a certain material causes uncertainties in release dates of the operations that require the material for its production. The main idea behind the approach is to assess whether the loaded operations of an incoming order can really commence its production in the periods where they are loaded. We refer to this as the feasibility of a solution. The way that we load the operations within the incoming order can have an impact on the feasibility a solution. Preventing to load operations that require materials with stochastic external lead times in periods that precedes the uncertain release dates can mitigate inventory shortages and not meeting due dates as a result. For the process of assessing the feasibility of solutions, we use the idea of sample average approximation (SAA) from Shapiro (2003) that we identified from the literature in Section 3.4.1.

We consider set V (index l), consisting of materials with stochastic external lead times. We draw scenarios ω_l from Ω_l for the external lead time of material l. More specifically, each scenario ω_l results in an external lead time $L_l(\omega_l)$, which results in a release date $r_j(\omega_l)$. The set Ω_l consists of (historical) data regarding external lead times of raw material l. As we aim to seek a for robust solution, we first determine the external lead times of all raw materials l based on the kth percentile from the data. For instance, we can use a 70th, 80th or 90th percentile, where higher percentiles would imply more robust release dates but can sometimes result in loading operations in unnecessary late periods. Afterwards, we construct a solution based on the following release dates. Finally, we randomly select a couple of times a scenario for each material from Ω_l to obtain the service level λ related to the feasibility of the solution. Figure 4.7 visualizes an overview of these steps.

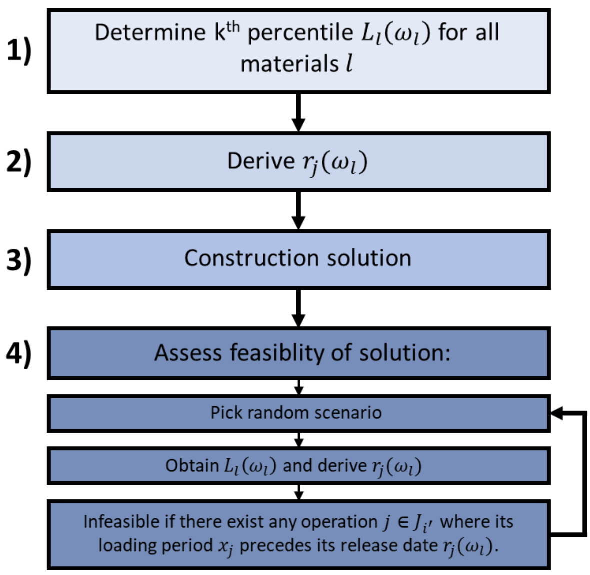

Figure 4.7; Steps to cope with stochastic external lead times

Below, we provide an overview of the new sets and parameters that we introduce within the stochastic formulation of the model. Besides, we elaborate on way in which we obtain the release dates $r_j(\omega_l)$ from the external lead times $L_l(\omega_l)$ as well as how we find the service level λ.

Set	Description
V	Set of materials with stochastic external lead times considered indexed by l
Ω_l	Set of all possible external lead time scenarios for raw material l indexed by ω_l, and $\Omega_l = \{1, \dots, \lvert\Omega_l\rvert\}$

Parameter	Description
$L_l(\omega_l)$	External lead time of material of type l under scenario ω_l
$r_j(\omega_l)$	Release date of operation $j \in J_{i'}$ under scenario ω_l
λ	Service level
RP_l	Receiving period of material l
RO_l	Number of periods passed since last replenishment order from raw material l

For a single scenario ω_l, we determine the release date of operation $j \in J_{i'}$ that requires one or multiple materials from set V as follows:

Period of receiving material l	Situation
$RP_l(\omega_l) = 0$	If operation j does not require material l
$RP_l(\omega_l) = 0$	If material l is on stock
$RP_l(\omega_l) = L_l(\omega_l),$	If material l still needs to be ordered
$RP_l(\omega_l) = L_l(\omega_l) - RO_l$	If materials l is already ordered RO_l periods ago

Eventually, the release date of operation $j \in J_{i'}$ becomes the moment when the last material arrives:

$$r_j(\omega_l) = \max_l\{RP_l(\omega_l)\}$$

To assess the feasibility of a solution and to obtain the service level λ (Step 4 in Figure 4.7), we first elaborate on what the service level implies. In our problem setting, the service level λ relates to the fraction of orders containing operations that require materials with stochastic external lead times that do not meet the due date d_i that follows the model in Section 4.3.

To obtain the service level λ, we first count the number of scenarios for which the solution is feasible by means of auxiliary variable $\alpha(t)$. We initialize $\alpha(t)$ to zero and add one for every scenario that results in a feasible solution in the following way:

$$\alpha(t) = \begin{cases} \alpha(t) + 1, & if\ all\ x_j \geq r_j(\omega_l) \\ \alpha(t) + 0, & if\ any\ x_j < r_j(\omega_l) \end{cases}$$

Eventually, the service level λ becomes the number of scenarios for which the solution is feasible divided by the total number of scenarios applied α^*:

$$\lambda = \frac{\alpha(t)}{\alpha^*}$$

4.5 Model approach conclusion

In this chapter, we propose a model capable of assigning due dates of incoming orders. The model does this by loading workloads, i.e., required production capacity, of orders into certain time periods at the corresponding production departments. For loading the workloads, we use a periodic loading framework at each production department, as mentioned in Section 4.1, where each time period has a certain maximum capacity. To optimize due date assignments, we allow the model to spread the workloads of orders over multiple periods, impose lateness to the already assigned due dates of existing orders and make use of nonregular production capacity by means of working in overtime.

In Section 4.3, we present the deterministic model formulation in a setting where an incoming order request arrives with a customer-requested due date when there are a set of existing orders with already assigned due dates that have not been produced yet. The model objective is to minimize the costs related to: 1) earliness or tardiness on the customer-requested due date, 2) earliness or tardiness on already assigned due dates of existing orders, 3) spreading the workload of a single order at a department over multiple period and 4) nonregular capacity usage.

At last, in Section 4.4, we extend the deterministic model to cope with stochastic external lead times that affect can the release dates of some operations within the incoming order. Here, we use the principle of sample average approximation (SAA) to assess the feasibility of a solution.

5. Solution design

This chapter outlines the approaches in which we load workloads of orders into the periodic loading framework. The purpose of these approaches is to generate an as good as possible solution in a competent time at the moment of a new order arrival. This implies fitting the workload of an incoming order in available capacity, which can provide us information regarding tactical level decisions such as assigning due dates and managing production capacities. To realize this, we use an approximate approach where we apply heuristics to come to an as good as possible solution. The reason for this approach is that it can obtain good quality solutions in a computationally efficient manner, making the approach applicable for practical solution generation. More specifically, heuristics enable decision-making and problem-solving in our setting where the focus is on finding satisfactory solutions within the available resources and time constraints rather than finding absolute optimality. Our solution design considers the use of construction heuristics to create an initial solution. Afterwards, we apply improvement heuristics to the initial solution to assess whether refinements can be achieved in satisfactory computational times.

Figure 5.1 depicts an overview of the solution design of this research, where the first step is to create an initial solution by means of construction heuristics. For this step, information regarding the incoming order (e.g. workloads or release dates), existing orders that have not been completed yet and production capacities are essential input. After we create an initial solution, the next step is to apply improvement heuristics to improve on the initially created solution. Note that it might be possible that certain construction heuristics perform well on their own within remarkably less computational time in comparison with including an improvement heuristic. That is why we apply an experimental design to assess the performance of the approaches in the setting of TOMRA. Chapter 6 elaborates further on this topic.

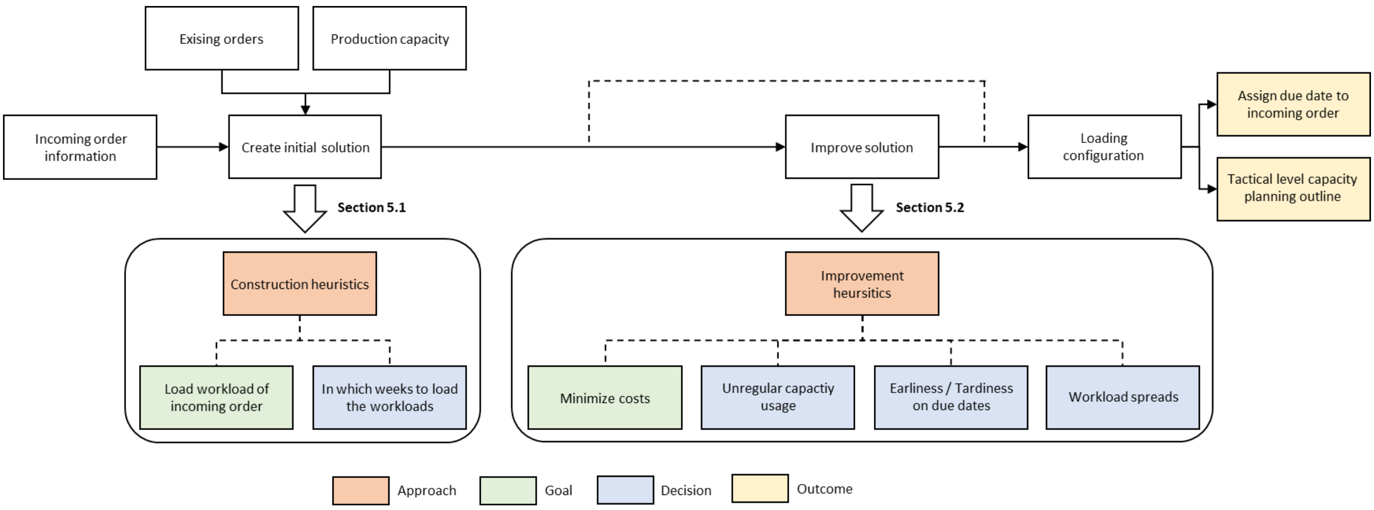

Figure 5.1; Solution design overview

Section 5.1 describes several construction heuristics that can create an initial solution. These approaches consist of ways we can load the workload of an incoming order with respect to orders that are already in the system. The approaches that we discuss within this section consist of solely loading the workloads of an incoming order as well as reloading the workloads of orders that have not been completed yet. Section 5.2 discusses improvement heuristics capable of adjusting a solution instance in search for the optimal solution. Finally, Section 5.3 provides a conclusion to this chapter.

5.1 Loading of incoming order
This section highlights several approaches to load the workload of an incoming order whilst regarding already loaded orders that have not been completed yet. These approaches act as construction heuristics to create an initial solution. We address a total of six approaches based on the findings related to finite loading methods within the literature of Section 3.2.3. The approaches are forward loading (FL), collective forward loading (CFL), hybrid loading (HL), unloading and forward loading (UFL), unloading and collective loading (UCL) and unloading and hybrid loading (UHL). We elaborate on each of these approaches in the corresponding subsections below.

5.1.1 Forward loading approach
The forward loading approach (FL) loads the workloads of an incoming order per department without adjusting or reloading existing orders that are already loaded during their arrival and have not been completed yet.

In summary, the FL approach tries to load the individual operations within the incoming order starting in the week of its release date. The literature in Chapter 3 addresses this way of loading as forward loading. We apply forward loading of the operations within the incoming order, starting with the operations that require production at the first department. There is no clear logic regarding which department to load first, since all operations require production at a single production department, making the departments independent of each other when it comes to loading the operations.

In our approach, we load an operation in a period if there is any regular capacity left within this week, even if some nonregular capacity will be used to load its entire workload together with the existing workload in the same week. The main reason for this way of loading is to occupy as much regular capacity as possible, which can contribute to assigning earlier due dates. Needless to say, if there is no regular capacity left or loading the operation results in exceeding the maximum overtime within a week, then FL looks to load the operation in the next week until the workload is loaded.

The FL approach does this way of loading for every single operation within the incoming order, where the FL approach loads the operation with the earliest release date first to take advantage of possible early release dates of operations. If there are operations with equal release dates, we aim to load the operation with the shortest processing time first. The reasoning behind this way of loading the operations is that this, in combination with loading operation when there is any regular capacity left, can lead to more operations to be loaded in earlier weeks, thus providing earlier due dates. Eventually, the FL approach loads the entire workload of the incoming order and a feasible solution is established. From this solution, amongst other things, a due date of the incoming order follows resulting in an objective value.

In addition, after all the operations of an order are loaded, we include an approach to check whether it is more beneficial to use extra nonregular capacity in earlier weeks to prevent tardiness. We elaborate on this approach later within this subsection.

We depict an overview on way the FL approach works by means of the pseudo code seen in Figure 5.2.

1) $x_j = z_j$ for all operations within existing orders ($i \neq i'$)
2) For all departments $s \in S$:
3) Determine total existing workloads per period $t \in T$ (WL_{st})
4) For all operations within $J_{i'}$ that require production at s ($y_j = s$):
4.1) Select operation with earliest release date
4.1.1) Operations with equal release dates?
Select operation with shortest processing time
4.1.2) Operations with equal release dates and equal processing times?
Select first operation within $J_{i'}$
4.2) Week $= r_j$
4.3) No regular capacity available in Week? ($WL_{st} > Q_{st}$, where $t =$ Week)
If yes: Week = Week + 1 (Go to 4.3)
4.4) Maximum overtime exceeded? ($WL_{st} + p_j > Q_{st} + MO_{st}$, where $t =$ Week)
If yes: Week = Week + 1 (Go to 4.3)
Otherwise: $x_j =$ Week (Go to 4.5)
4.5) Update total workloads ($WL_{st} = WL_{st} + p_j$)
5) Decisions extra nonregular capacity to avoid tardiness

Figure 5.2; FL approach procedure

Next, we provide a simplified example illustrating the way in which the FL approach creates a solution.

Figure 5.3 depicts the workload of an incoming order at a certain department. Within this example, this workload consists of 5 individual operations j, named 1 through 5. The workloads of these operations should be loaded within certain weeks at this department. Each of these operations has a release date r_j indicating the first week in which an operation can be loaded. These release dates are input parameters and can in some instances, namely for operations requiring materials with stochastic external lead times, be derived with the approach from Section 4.4.

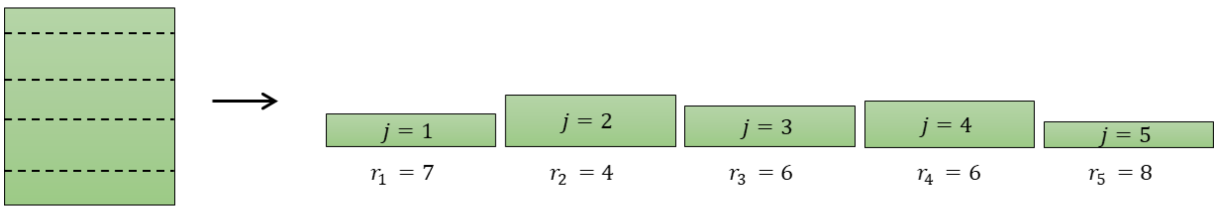

Figure 5.3; Release dates of operations within incoming order example

Next to the input of release dates of the operations, orders from which the workloads are already loaded at a certain department during their earlier arrival are also input. Figure 5.4 provides an overview of the workloads of the existing orders that have not been completed yet. We represent these loaded workloads in the periodic loading framework, with a certain regular capacity Q_{st} and maximum overtime capacity MO_{st} per week.

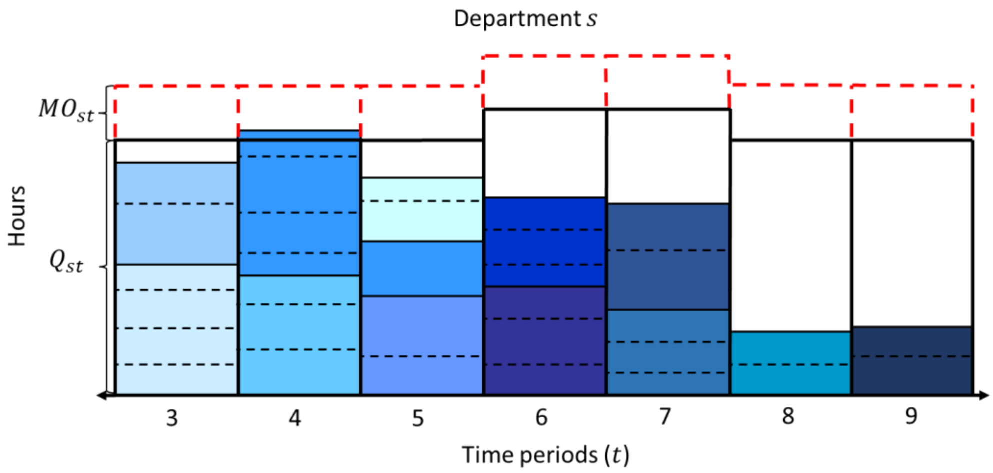

Figure 5.4; Example of periodic loading framework with existing workloads

Within this figure, each color represents a different order, where we depict individual operations within the orders as well. Note that the operations of each existing order also have their individual release dates, however since FL does not consider reloading these operations, these are not relevant to mention here.

Figure 5.5 visualizes the way that FL loads the workloads of the operations within the incoming order. Operation 2 of the incoming order has the earliest release date, namely week 4, however the existing workloads within this week already exceed the maximum regular capacity. Because of this, we look at the next week where there is regular capacity left. This results in loading this operation 2 within week 5. All other operations within the incoming order can be loaded in the same week as their release date in this situation.

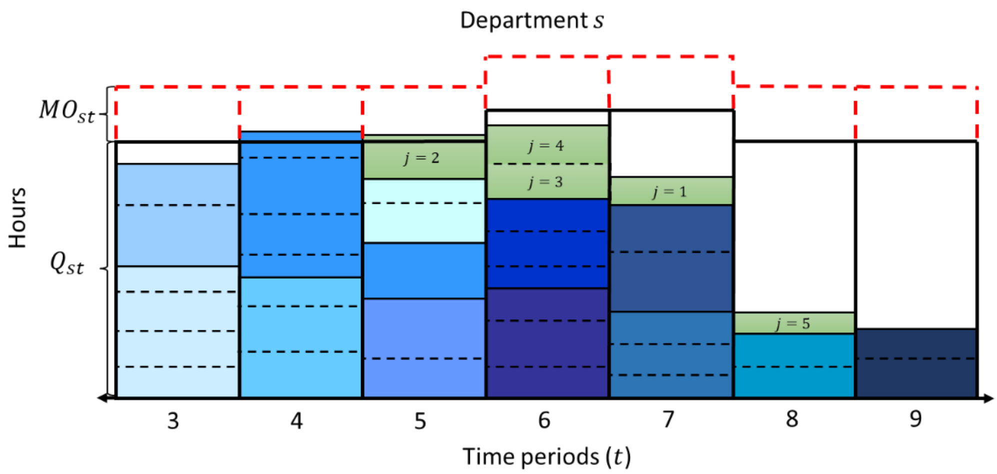

Figure 5.5; FL approach example

The basic idea behind the FL approach is that it tries to load operations when there is remaining regular capacity left within a week. This can result in the occupation of remaining capacity, making sure that regular capacity is well utilized. On the other side, the workloads of orders can be spread out over a large number of periods when using the FL approach. This can happen if the release dates of the individual operations differ (as within the example) or if multiple weeks after the release dates have little capacity left where the FL approach can only load 1 operation per week for instance.

Moreover, the FL approach allows for using nonregular capacity as a result of loading an individual operation. Operation 2 that is loaded within week 5 occupies the remaining regular capacity during this week, but also results in some nonregular capacity usage which we allow to happen if loading its workload does result in exceeding the maximum overtime usage. This is because the use of nonregular capacity is not desired but can definitely be used in order to efficiently load the workloads of orders.

Figure 5.6 provides a new situation of loaded workloads in the periodic loading framework, where the existing workloads in weeks 6 and 8 are larger in comparison with the previous situation.

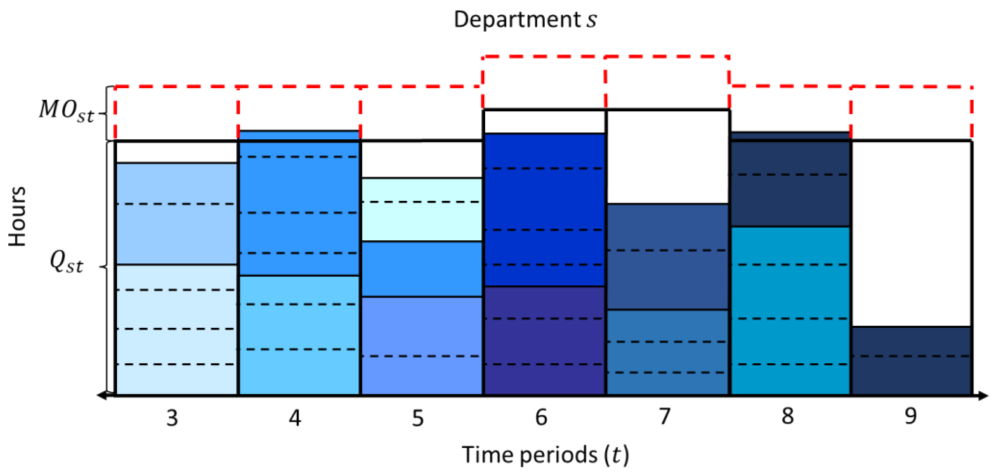

Figure 5.6; Example of periodic framework with more existing workloads

Figure 5.7 shows how the same incoming order is loaded in this new situation. Again, the FL method tries to load the operations where there is regular capacity available. Note that operation 4 is loaded one week later as a result of this new setting. Moreover, Figure 5.6 indicates the customer-requested due date of the incoming order ($cd_{i'}$), which is in week 9.

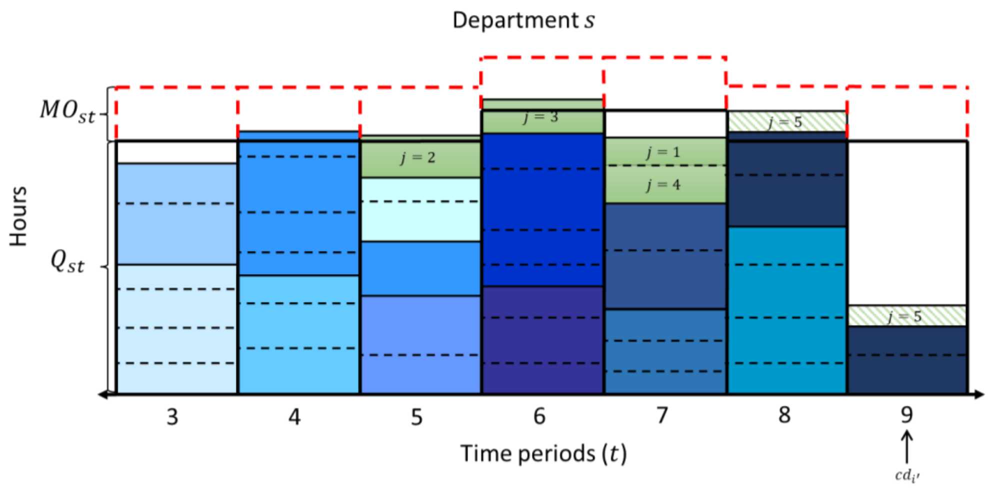

Figure 5.7; Decisions FL approach

As mentioned before, FL tries to load operations where there is regular capacity left, which should imply that we load operation 5 in week 9 within this example. However, when we make this decision, we might incorporate tardiness related the customer-requested due date. This is because one can include a certain slack of q weeks in assigning due dates, where the due date of the order equals q plus the week of last loaded operation (see Section 4.3). In the case where $q = 1$, the assigned due date would be in week 10, causing one week tardiness on the customer-requested due date. Therefore, it might be beneficial to load operation 5 in week 8, using some nonregular capacity whilst meeting $cd_{i'}$.

We refer to the operations that determine the due date of an order with tardiness as bottleneck operations. Within the example where we consider only one production department, operation 5 is a bottleneck operation. Besides, if there are multiple production departments, the bottleneck operation is the last loaded operation of an order for all departments. The implies that a bottleneck operation determines the due date of the entire order, which, needless to say, can be multiple operations. Because of this, we consider using nonregular regular capacity in earlier weeks for just these bottleneck operations to try to reduce the tardiness of an order.

As it always favorable to assess whether is it more beneficial to use some nonregular capacity for the bottleneck operations to prevent tardiness, we include an approach that does this. The last step of the FL approach explained in Figure 5.2 relates to this approach.

We only consider these decisions when the due date of the initial solution, that follows from the previous steps of the FL approach, exceeds the customer requested due date ($d_{i'} > cd_{i'}$). There are multiple ways in which one is able to prevent tardiness, where we aim to design a relatively simple procedure to make these decisions instead of designing an extensive improvement heuristic for this. In our approach, we keep loading the bottleneck operations one week earlier starting with the bottleneck operation that has the latest release date if there are multiple. The reason behind selecting the operation with the latest release date first is that it is more likely for bottleneck operations to be loaded in weeks that preceded their release date otherwise. Moreover, as the approach would always use nonregular capacity to prevent tardiness, we do not check whether it is more beneficial to load the bottleneck operations more than one week ahead in a single iteration. The procedure stops when operations cannot be loaded in an earlier week or when there is no tardiness anymore. Eventually the procedure returns the best found solution, which does not necessary have to be solution without tardiness.

In Figure 5.8, we provide pseudo code that explains the logic behind the decisions to use extra nonregular capacity to prevent tardiness, which acts as the final step of our FL approach.

1) Best solution = Initial solution
   Current solution = Initial solution
2) Does incoming order have tardiness? ($d_{i'} > cd_{i'}$)
   If yes: Go to 3
   Otherwise: Go to 7
3) Determine all bottleneck operations of the order in the current solution ($j|x_j = \max_s\{c_{i's}\}$)
4) For all bottleneck operations, do the following:
   4.1) Select bottleneck operation with latest release date
      4.1.1) Operations with equal release dates?
         Select operation with shortest processing time
      4.1.2) Operations with equal release dates and equal processing times?
         Select first operation within $J_{i'}$
   4.2) Release date of operation equal to loading period? ($r_j = x_j$)
      If yes: Go to 7
   4.3) Load operation 1 week earlier ($x_j = x_j - 1$)
   4.4) Maximum nonregular capacity exceeded? ($WL_{st} + p_j > Q_{st} + MO_{st}$)
      If yes: Go to 4.2
   4.5) Update total workloads ($WL_{st} = WL_{st} + p_j$)
5) Obtain new solution
6) Objective new solution < Best solution
   If yes: Current solution = New solution
   Best solution = New solution (Go to 2)
   Otherwise: Current solution = New solution (Go to 2)
7) Result = Best solution

Figure 5.8; Procedure decision regarding extra overtime usage to prevent tardiness of incoming order

We use the logic explained within the Figure 5.8 to balance between earlier due date assignment and nonregular capacity usage for the other loading approaches that we discuss later within this section as well.

5.1.2 Collective forward loading approach

Just like the FL approach, the collective forward loading approach (CFL) also loads the workload of an incoming order without reloading the existing orders that have not been completed yet.

In contrast with the FL approach, the CFL approach is a self-constructed approach and loads the operations within the incoming order at a department from the moment when all operations can begin their production at this department onwards. This stimulates the operations that require production at a certain department of an incoming order to be loaded in periods at close proximity. We denote the latest release date of any operation j within incoming order i' that requires production at a certain department s as the collective loading period at this department. We formulate this collective loading period as follows:

$$Collective\ loading\ period_s = \max_{j \in J_{i'} | y_j = s} \{r_j\}$$

Appendix B contains an overview the logic behind this approach as well as the other approaches that we discuss during this Section.

Figure 5.9 shows an example of the solution instance resulting from the CFL approach. The setting of this example is the same as the example from Section 5.1.1, where the operation of an incoming order (Figure 5.3) should be loaded into a periodic framework with already loaded orders (Figure 5.4). In this example, operation 5 has the latest release date, namely in week 8. So, the CFL approach loads all the operations from the incoming order at this department from week 8 onwards. Within this example, all five operations can be loaded into week 8.

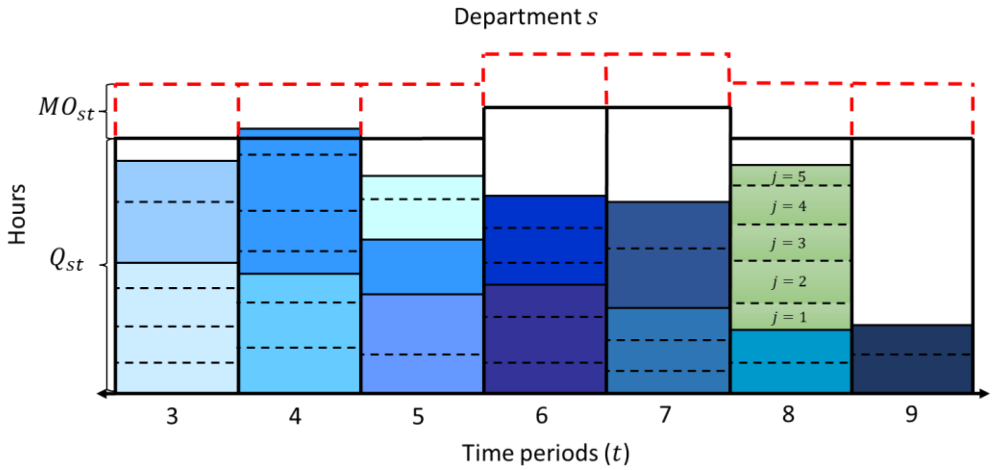

Figure 5.9; CFL approach example

Since the CFL approach waits with loading the operations at a department until all operations can begin its production, we expect the operations not to be loaded in a large number of different time periods. However, this can result in a later completion of all operations at a department. For instance, if the existing workloads of week 8 in the example of Figure 5.10 were larger, then some operations of the incoming order should be loaded in week 9.

Just like within the FL approach, the CFL approach allows the usage of nonregular capacity during loading the operations. Operations of the incoming order will occupy the remaining capacity from the collective loading period until the full maximum regular capacity is occupied. This implies that if there is regular capacity available and the workload of an operation exceeds this available regular capacity, then nonregular capacity is used. In addition, the CFL approach applies the same principle to balance between tardiness related to the customer-requested due date and nonregular capacity usage to prevent this tardiness as explained in Figure 5.8.

5.1.3 Hybrid loading approach

The hybrid loading approach (HL) is another approach that we developed and loads the workload of an incoming order without moving the existing orders that have not been completed yet. The HL approach uses both forward and backward loading, making it a hybrid approach. As mentioned before, forward loading loads the workloads of operations within the incoming order from the week of their release dates onwards. Backward loading loads the workloads of operations the other way around, starting with loading the operations in the week that results in the customer requested due date ($cd_{i'}$) and loading in the preceding weeks if the regular capacity limit is reached. The reason for making a hybrid approach instead of applying solely backward loading is that we do not always impose earliness costs in our problem setting. More specifically, some orders or departments do desire to complete the production activities ahead in time, which is something that principle of backward loading tends to counteract.

In summary, the choice for loading the operations j of incoming order i' forward or backwards depends on two factors, namely the cost of preceding the customer-requested due date ($\mathrm{RCE}_{i'}$) and the costs of completing production activities early at a department (ACE_s). Table 5.1 provides an overview of when the HL approach applies forward or backward loading based on these two factors.

Table 5.1; HL approach guidelines

		Department s where:	
		$\mathrm{ACE}_s > 0$	$\mathrm{ACE}_s \leq 0$
Incoming order i' with:	$\mathrm{RCE}_{i'} > 0$	Backward loading	Forward loading (CFL)
	$\mathrm{RCE}_{i'} \leq 0$	Forward loading (CFL)	Forward loading (CFL)

If $\mathrm{RCE}_{i'} > 0$, then the customer with the incoming order does not desire an earlier due date than its requested due date. When this is the case, we allow the operations to be loaded backwards, as we aim to avoid deviation from the customer-requested due date. However, we only allow this backward way of loading to happen at the departments that do not desire early production, i.e., the departments where $\mathrm{ACE}_s > 0$.

The process of backward loading goes as follows: We load the operations within the incoming order that require production at these departments from period $cd_{i'} - q$ backwards. We include slack q, since this determines the due date that we assign to the incoming order. If there is any regular capacity left at these departments within period $cd_{i'} - q$, then we load the operations at that department, starting with the operations that has the latest release date, until all regular capacity is occupied. If there is no regular capacity left, then this backward loading looks at the previous week until all operations are loaded. If now an operation is loaded in a week that precedes its release date, then we start again with backward loading but now from week $cd_{i'} - q + 1$. This extends the due date with one week, causing tardiness on the customer-requested due date. This is applicable if there is not enough regular capacity available or when $cd_{i'} - q$ automatically precedes one or multiple release dates, making the requested due date infeasible.

For departments that do desire to be finished early with the production of order ($\mathrm{ACE}_s \leq 0$), we apply forward loading. The reason behind this is that backward loading generally results in loading workloads in later periods in comparison with forward loading, since this way of loading looks to loads the workloads in available capacity close to the customer-requested due date. This implies that backwards loading results in less earliness, however if a department has benefits for earliness than we apply forward loading here.

In the case when $\mathrm{RCE}_{\mathrm{i}'} \leq 0$, then the customer with the incoming order does desire an earlier due date than its requested due date or a due date as soon as possible. When this happens, the model includes benefits for assigning earlier due dates. The HL approach loads the operations of incoming order i' at all departments by means of forward loading when $\mathrm{RCE}_{\mathrm{i}'} \leq 0$.

When applying forward loading during the HL approach, we choose to use the CFL approach instead of the FL approach. The reason behind this is that the backward loading technique that we apply tries to load all the operations at a department within the same period, namely period $cd_{i'} - q$. The CFL approach also tries to load the operations at a department in the same period, namely the collective loading period.

Figure 5.10 provides pseudo code that explains the logic behind the HL approach.

```
1)  x_j = z_j for all operations within existing orders (i ≠ i')
2)  Incoming order requests compliance with due date? (RCE_i' > 0)
        If yes:         Go to 3
        Otherwise:      Go to 8

3)  For all departments s ∈ S:
4)  |   Determine total existing workloads per period t ∈ T (WL_st)
5)  |   Does department desires early completion? (ACE_s > 0)
    |           If yes:                 Go to 6
    |           Otherwise:              Go to 7
    |
6)  |   For all operations within J_i' that require production at s (y_j = s):
    |   |       6.1) Select operation with latest release date
    |   |               6.1.1) Operations with equal release dates?
    |   |                       Select operation with shortest processing time
    |   |               6.1.2) Operations with equal release dates and equal processing times?
    |   |                       Select first operation within J_i'
    |   |       6.2) Counter = 0
    |   |       6.3) Week = cd_i' − q
    |   |       6.4) Counter = Counter + 1
    |   |       6.5) r_j > Week?
    |   |               If yes:         Week = cd_i' − q + Counter (Go to 6.4)
    |   |               Otherwise:      Go to 6.6
    |   |       6.6) No regular capacity available in Week? (WL_st > Q_st, where t = Week)
    |   |               If yes:         Week = Week - 1         (Go to 6.5)
    |   |       6.7) Maximum overtime exceeded? (WL_st + p_j > Q_st + MO_st, where t = Week)
    |   |               If yes:         Week = Week - 1         (Go to 6.5)
    |   |               Otherwise:      x_j = Week              (Go to 6.8)
    |   |       6.8) Update total workloads (WL_st = WL_st + p_j)
7)      For all operations within J_i' that require production at s (y_j = s):
        |       CFL Approach

8)  For all departments s ∈ S:
9)  |   For all operations within J_i' that require production at s (y_j = s):
    |   |       CFL Approach

10) Decisions extra nonregular capacity to avoid tardiness
```

Figure 5.10; HL approach procedure

Next, we provide a simplified example of way the HL approach applies backward loading. Figure 5.11 provides the workloads of an incoming order at a certain department, consisting of four operations each with their own release dates. Moreover, the figure specifies the customer-requested due date of the incoming and the customer does not desire an earlier due date ($\mathrm{RCE}_{\mathrm{i}'} > 0$).

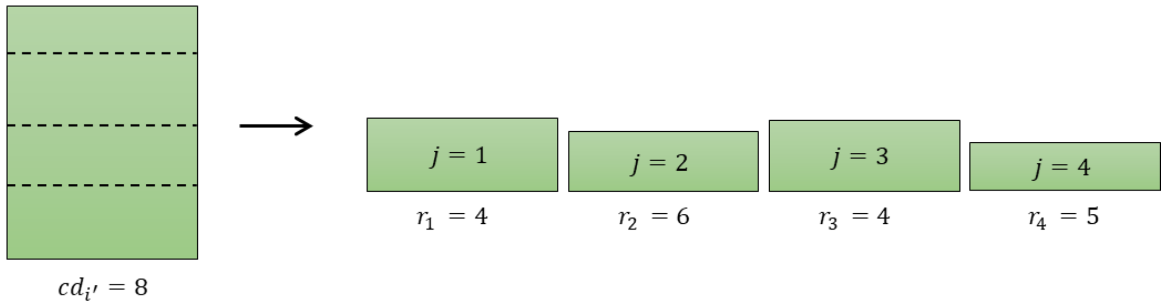

Figure 5.11; Example of customer-requested due date of incoming order and release dates of its operations, where $RCE_{i'} > 0$

Additionally, Figure 5.12 shows the loaded workloads of the existing orders at the department in question. Each color represents the workload of a different order, where the individual operations are not visible. Moreover, this department does not desire early production, meaning that $\mathrm{ACE}_{\mathrm{s}}$ of this department is larger than zero.

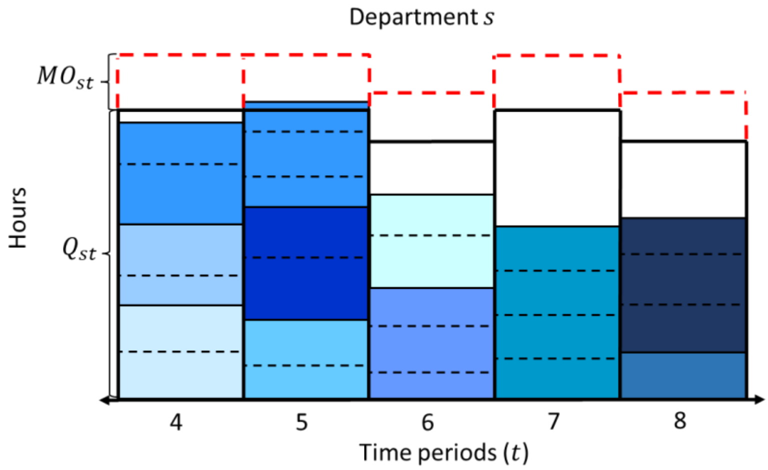

Figure 5.12; New example of periodic loading framework with existing workloads of department s where $ACE_s > 0$

In this example, the slack for assigning due dates is one week ($q = 1$). Next, we illustrate how the operations within the incoming order can be loaded with respect to available capacities at this department. Since both $\mathrm{RCE}_{\mathrm{i}'}$ and $\mathrm{ACE}_{\mathrm{s}}$ are positive, earliness is not desired and we apply backward loading. Figure 5.13 visualizes the way that HL loads the workloads of the incoming order into the periodic loading framework. $cd_{i'}$ equals 8 and q equals 1, this implies that we can start with loading in week 7 ($cd_{i'} - q$). We can observe that there is still regular capacity available within this week, so we can load (some of) the operations in week 7. Within Figure 5.13, one can observe how HL applies backwards loading in this setting. We first load the operation with the latest release date, which is operation 2. Afterwards, we fill the available regular capacity until all regular capacity is occupied. Afterwards, we look one week back in time, where there is still regular capacity left to load the remaining workload of the incoming order at this department.

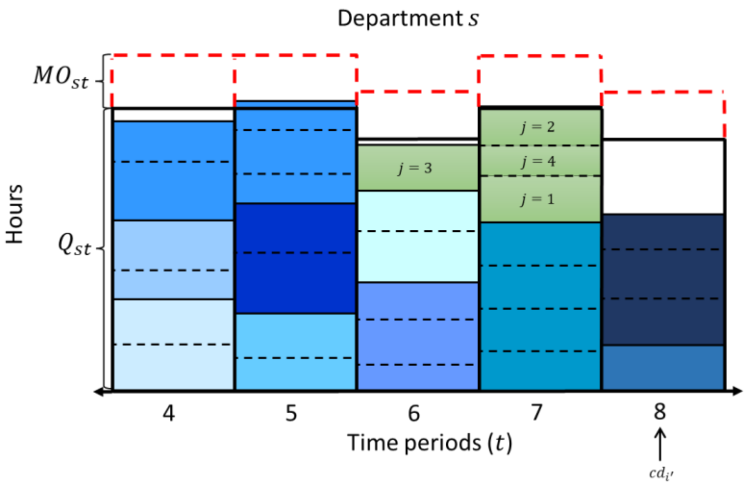

Figure 5.13; HL approach example in situation where $RCE_{i'} > 0$ and $q = 1$ at department s where $ACE_s > 0$

Within the example, not all operations can be loaded within the regular capacity of period 7. That is why we load operation 3 in period 6 instead based on the procedure from Figure 5.10. Furthermore, note that if the existing workloads in weeks 4 through 7 in the example were larger, then there would not be enough available regular capacity to load all the operations before the customer-requested due date. In this situation, (some of) the operations would have been loaded in week 8, causing tardiness on the customer-requested due date with a positive slack. Note that this could also have been the case when the release dates of all operations were in week seven or later in the current example.

Since the HL approach considers desired or undesired early production, the solutions that follow from this approach most likely have low costs on this aspect. On the other hand, the way of backward loading that the HL approach proposes can result in the spread of workloads over a large number of weeks. This can be the case if there is little regular capacity left in multiple weeks, where only one or a few operations can be loaded at a time.

Again, just like with the other approaches, the HL approach allows the usage of nonregular capacity. Initially, HL generates an initial solution whilst allowing only loading operations if there is regular capacity available. Moreover, HL uses the same logic from Figure 5.8 to balance between nonregular capacity usage and preventing tardiness.

5.1.4 Unloading and forward loading approach

The unloading and forward loading approach (UFL) loads the workloads of an incoming order in a similar manner to the FL approach, however the UFL approach considers adjusting or reloading the workloads of existing orders that have not been completed yet.

The UFL approach iteratively loads all order operations of orders with assigned due dates outside the frozen periods ($ad_i > |H|$) in the same manner as the FL approach. This implies that we first 'unload' existing workloads of orders with an already assigned due date outside the frozen periods, after which we load them again. In our approach, we unload all workloads of orders with a due date outside the frozen periods, however an option could also be to unload a fraction of these workloads for computational efficiency. The main reason for still unloading all the workloads in our approach is that all loaded workloads take each other into account. For instance, loading some workloads in a different manner could free regular capacity and could have impact on all succeeding loaded workloads.

When it comes to loading, we start with loading operations from the order with the earliest due date, with the reason to meet the already assigned or requested due dates. We select the order with the shortest total processing time if there are multiple orders with equal due dates. This is because, this way, more orders can potentially be loaded in earlier time periods. Needless to say, in our approach, we load the workloads of one order after the other. However, one could, for instance, decide to load all the workloads per department (one department after the other). The reason for our approach to iteratively load the orders instead is to ensure coordination between the workloads of individual orders at all production departments.

Moreover, the UFL approach also balances between nonregular capacity usage and tardiness of either the customer-requested due date of the incoming order or the already assigned due dates of existing orders. The logic that Figure 5.8 depicts also accounts for the UFL approach. The only adaptation to this procedure is that it should also be executed for existing orders that are reloaded and experience tardiness related to their already assigned due date ($d_i > ad_i$). In this situation, the procedure should check whether it is worth using some nonregular capacity to prevent the costs of tardiness related to an already assigned due date.

Figure 5.14 provides an overview of the logic behind UFL approach by means of pseudo code.

1) For all orders $i \in N$ do the following:
2) Select order with earliest due date (ad_i or $cd_{i'}$)
   2.1) Orders with equal due dates?
      Select order with shortest total processing time (workload)
   2.2) Orders with equal due dates and equal workloads?
      Select first order within N
3) Does selected order have a due date within a frozen period? ($ad_i \leq |H|$)
   If yes: Do not change loading periods of order ($x_j = z_j \quad \forall j \in J_i$)
   Update total workloads ($WL_{st} = WL_{st} + p_j$) & Go to 2
   Otherwise: Go to 4
4) For all departments $s \in S$:
5) For all operations within J_i that require production at s ($y_j = s$):
   FL approach
6) Decisions extra nonregular capacity to avoid tardiness

Figure 5.14; UFL approach procedure

Next, we provide a simplified example on the way in which the UFL approach creates a solution at the moment of a new order arrival. Within this example, we only consider one production department and have a slack of one week for assigning due dates ($q = 1$).

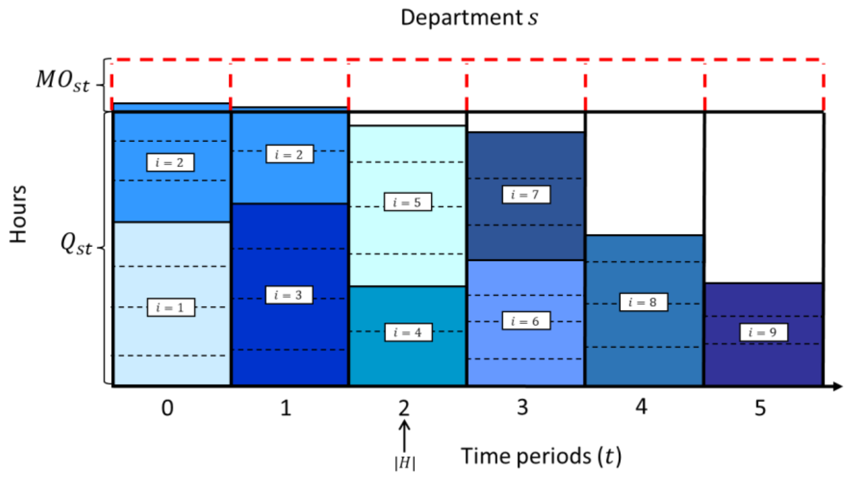

Figure 5.15; Another example of periodic framework with existing workloads of a department

Figure 5.15 provides an overview of the loaded workloads of existing orders at the moment of a new order arrival. In total there are nine existing orders ($i = 1, \dots, 9$). Moreover, the first three weeks, starting with week 0, make up the frozen periods, indicating that $|H|$ equals 2.

Furthermore, Figure 5.16 provides information regarding the incoming order (index $i = 10$). This incoming order has a customer-requested due date in week 6 and consist of four operations, each with their own release date.

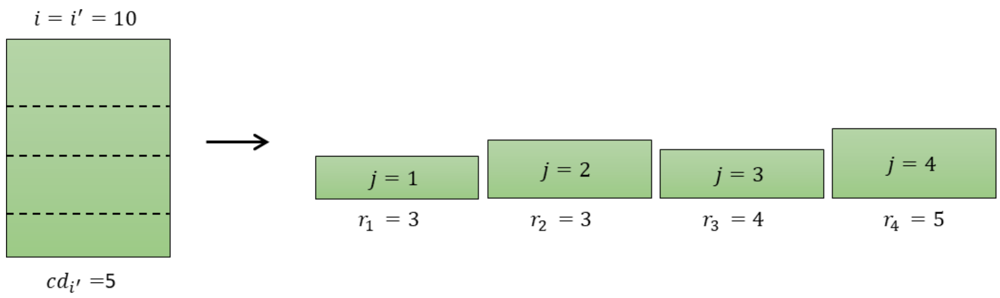

Figure 5.16; Incoming order with its operations and customer-requested due date example

Table 5.2 gives the last part of the input, showing the due dates of all the orders within the system. These due dates are either already assigned due dates of existing orders (ad_i) or the customer-requested due date of the incoming order ($cd_{i'}$).

Table 5.2; Due dates of orders example

Orders $i \in N$	ad_i / $cd_{i'}$
1	1
2	2
3	2
4	3
5	4
6	4
7	4
8	5
9	6
10	5

After we defined all of the input, we can start with unloading the workloads of orders with a due date outside the frozen periods. Orders 1, 2 and 3 all have a due date within a frozen period with the current slack, meaning that we do not unload the workloads of these orders. Figure 5.17 provides an overview the periodic loading framework after we unloaded the corresponding workloads.

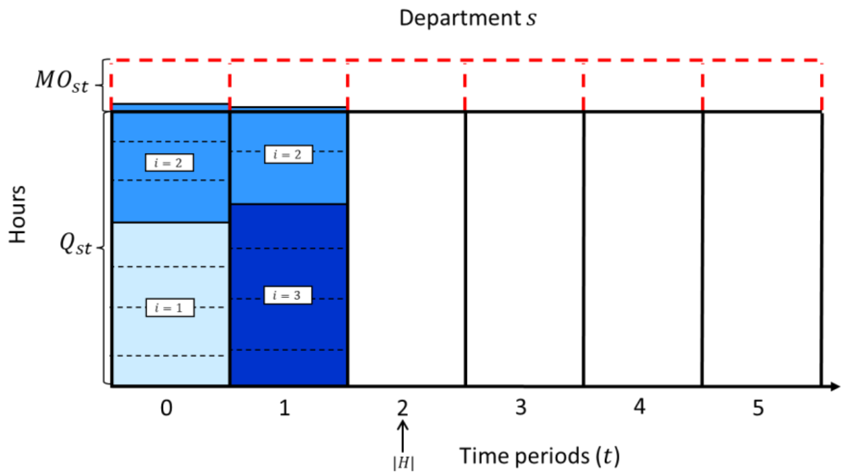

Figure 5.17; Periodic framework after unloading workloads of orders with $ad_i > |H|$

After unloading the workloads, the UFL approach iteratively starts with loading the operations of orders, beginning with the order that has the earliest due date. Figure 5.18 provides the solution instance that follows after reloading the workloads according to the UFL approach. Note that we do not provide details regarding all operations of the existing orders as well. The UFL approach reloads the workloads of orders 4 through 8 before the incoming order (10). This implies that there was still capacity available in week 4, where we load the operation with the earliest release date of the incoming order. The remaining operations within the incoming order are loaded in the fifth week, as there was still available capacity within that period.

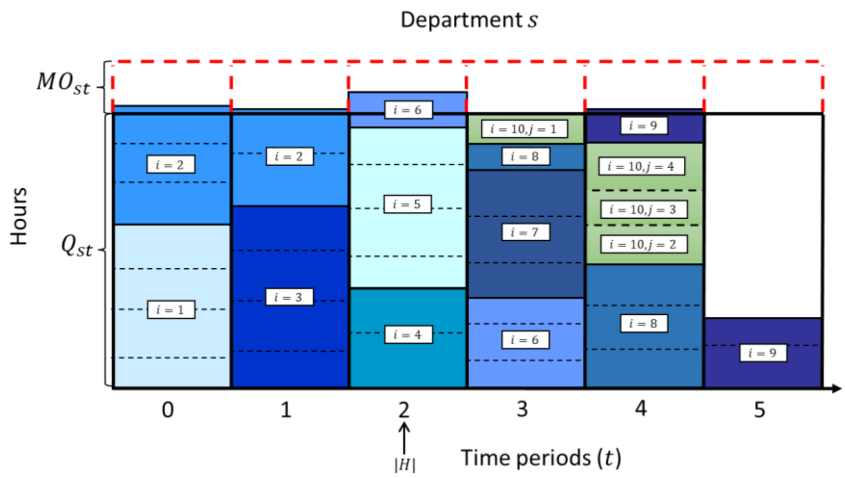

Figure 5.18; UFL approach example

Because of the possibility to reload workloads of existing orders combined with forward loading the operations separately in remaining regular capacity, the solutions following from the UFL approach can result in well-occupied regular capacities across the time horizon. However, just like the FL approach, the UFL approach can result in the workloads from single orders to be spread over a large number of weeks.

5.1.5 Unloading and collective loading approach

The fifth approach, namely the unloading and collective loading approach (UCL), also can adjust or reload the workloads of orders that have not been completed yet to load the workload of the incoming order.

Just like the UFL approach, the UCL approach first unloads the workloads of the existing orders with a due date outside the frozen periods ($ad_i > |H|$). Afterwards, the UCL approach iteratively loads the operations of orders, starting with the order that has the earliest due date (either ad_i or $cd_{i'}$). The UCL approach loads operations within orders that require production at a certain department in a forward manner from the collective loading period onwards, which is the same principle as the CFL approach.

After the workloads of an order have been loaded, the UCL approach, just like the other approaches, checks whether tardiness related to the due date ($d_i > ad_i$ or $d_i > cd_{i'}$) takes place and if it is more beneficial to use nonregular capacity to avoid this.

The way in which the UCL loads the workloads can result in less spread of the workloads of single order, as the approach tries to load all operations of order within the same period. On the other hand, this can result in less occupation of regular capacities in earlier weeks where some operations of an order could have been produced.

5.1.6 Unloading and hybrid loading approach

The final approach to load the workload of an incoming order is the unloading and hybrid loading approach (UHL). As the name implies, this approach is a combination between unloading existing orders and the previously discussed HL approach.

The UHL approach iteratively loads all order operations of orders with assigned due dates outside the frozen periods ($ad_i > |H|$) in the same manner as the HL approach. This implies that, again, the choice between forward or backward loading for the incoming order depends on both if the customer desires earliness ($RCE_{i'}$) as well as if the department desires earliness (ACE_s), see Table 5.1. For the existing orders with already assigned due dates, the way of loading only depends on ACE_s.

After the UHL approach loads the workloads of a single order, we check whether it is more beneficial to use nonregular capacity to avoid tardiness related to either the already assigned due date of existing order or the customer-requested due date of the incoming order.

5.2 Improving the initial solution

This section outlines approaches along the lines of improvement heuristics that can improve on initially created solutions. First, we elaborate on the choice of neighborhood operators that can construct neighborhood solutions. Afterwards, we discuss two improvement heuristics that we identified from the literature and we can apply to initially created solutions.

5.2.1 Neighborhood operators

For improving on initially created solutions, we select the following neighborhood operators based on the literature from Section 3.5:

Operator 1) Move operation j to a consecutive period at the same department s

Operator 2) Swap of operations j and j^* within consecutive periods at the same department s

Operator 3) Pairwise swap of all operations of orders i and i^* within consecutive periods at the same department s

Operator 4) Pairwise swap of all operations of order i that are loaded in the same period with an equal number of operations from other orders that are loaded in a consecutive period

We select operators 1-3, as these are commonly used neighborhood operators in a scheduling environment and are directly applicable in our problem setting. These three neighborhood operators allow changing the loading periods of operations by performing relatively simple actions at the same production department. Whilst operator 1 changes the loading period of a single operation, operators 2 and 3 interchange the loading periods of multiple operations to construct neighborhood solutions.

A disadvantage of neighborhood operators is that operators 1-3 cannot prevent tardiness of orders if there are bottleneck operations at different departments. We stress again that bottleneck operations are the latest-loaded operations within an order that cause tardiness on the due date. If we move or swap operations at one department at a time, then moving a bottleneck operation forward resulting in an earlier completion at a department can mean a worse solution if there is still a bottleneck operation at another department.

Neighborhood operator 4 can avoid tardiness, as it can load all bottleneck operations at every department of an order in the preceding period. This operator only applies if we select a bottleneck operation to be loaded in a different period. Note that operator 4 swaps the bottleneck operations at all departments with an equal number of operations instead of with operations from the same order. This is because, restricting a swap with only operations from the same order is not always feasible, since not every order must have a workload at every department.

Our strategy for selecting a neighborhood operator is that we select operators 1-3 with equal probabilities and select operator 4 in case that we decide to construct a neighborhood solution by changing the loading period of a bottleneck operation. The main reason for selecting operators 1-3 with equal probabilities instead of, for instance, selecting the best performing operator is to increase exploration among the solution space in search of the global optimum.

In the cases when we choose to apply neighborhood operators 2 or 3, we randomly select other one or multiple operations in a consecutive period relative to the first selected operation(s) to swap with. However, not all operations can be loaded in earlier or later periods, as they are restricted by release dates, or can cause tardiness otherwise. Table 5.3 indicates the actions that the loaded operations can perform in which situations.

Table 5.3; Actions of operations

In case of existing order	In case of incoming order	Release date	Action
$x_j < ad_i - q$	$x_j < cd_{i'} - q$	$x_j = r_j$	Move forward
$x_j < ad_i - q$	$x_j < cd_{i'} - q$	$x_j > r_j$	Move forward and backward
$x_j = ad_i - q$	$x_j = cd_{i'} - q$	$x_j = r_j$	Cannot move
$x_j = ad_i - q$	$x_j = cd_{i'} - q$	$x_j > r_j$	Move backward
$x_j > ad_i - q$	$x_j > cd_{i'} - q$	$x_j = r_j$	Cannot move
$x_j > ad_i - q$	$x_j > cd_{i'} - q$	$x_j > r_j$	Move backward

It is of great importance to construct feasible neighborhood solutions and the actions coupled to the situations stated in Table 5.3 help with ensuing feasible solutions. In particular, when using neighborhood operators 2 and 3, we should make sure that we swap operations that can be loaded in succeeding periods with operations that can be loaded in preceding periods. When using the third operator, for instance, and one of the operations j from selected order i that is loaded in a certain week cannot move, then we do not allow the neighborhood operator to continue. Additionally, the actions with the corresponding situations in Table 5.3 also restrict neighborhood solutions from causing more tardiness.

At last, one should always check whether an action results in exceeding the maximum overtime usage during a period at a department to enhance the feasibility of solutions.

All neighborhood operators (1-4) allow operations to be loaded in different periods at every department. This implies that the neighborhood operators can transform any initial solution at a department into another solution whilst considering the loading restrictions. Moreover, the neighborhood operators can do so within a finite number of iterations, as we do not impose any restrictions on the use of the operators apart from only allowing the use of operator 4 when we select a bottleneck operation. This makes that the neighborhood operators ensure a connected neighborhood.

5.2.2 Adapted steepest descent

The first improvement heuristic is an adaptation to the steepest descent heuristic that we identified from the literature and discuss in Section 3.5.2. For the steepest descent heuristic, we recall that the general idea is to find neighbor solution x' with the lowest objective function value $f(x')$ within the neighborhood $N(x)$ of initial solution x. We select this improvement heuristic, since it is capable of finding better solutions relatively fast (Pirlot, 1996). As it is not feasible to assess and construct every neighborhood solution in a timely manner and we still aim to find better solution fast, we adapt the steepest descent heuristic from Section 3.5.2 by only constructing neighborhood solutions for a fixed number of iterations. We construct these neighborhood solutions by randomly selecting operations with equal probabilities to apply a neighborhood operator to with the strategy explained in Section 5.2.1, apart from the operations of orders with an already assigned due date within the frozen periods ($ad_i \leq |H|$).

After we perform the iterations, we assess whether there is any improvement in comparison with the initial solution. If this is the case, then we continue with constructing neighborhood solutions, but now in the neighborhood of the best found solution for, again, a fixed number of iterations. This way, we constantly try to find the best neighbor until no improvements occur or a until we reach a certain time limit, where the latter is also an adaptation to the steepest descent heuristic. Figure 5.19 provides an overview of the procedure we use for the adapted steepest descent.

```
Generate initial solution x
Set x* = x
Set Stop = false

While Stop = false and Time < TimeLimit
    For i = 0 to MaxIterations
        x' = random solution in N(x)
        If f(x') < f(x*):
            x* = x'
        End
    End
    If f(x*) ≥ f(x)
        Stop = true
    Else
        x = x*
    End
End
Result = x*
```

Figure 5.19; Adapted steepest descent

5.2.3 Simulated annealing

The second improvement heuristic is the simulated annealing heuristic (SA) that we identified from the literature in Section 3.5.2. SA is a metaheuristic, meaning that, in contrast with steepest descent, it can escape from local optima (Henderson et al., 2003). The reason for selecting the SA heuristic instead of another metaheuristic like tabu search is that we do not go through the entire neighborhood, i.e., we do not construct and assess every possible neighborhood solution. Instead, we randomly construct neighborhood solutions in the same manner as described within the adapted steepest descent approach. From the literature in Section 3.5.2, it follows that SA always accept better or equally good neighborhood solutions x' (when $f(x') < f(x)$) and accepts worse solutions with a certain probability depending on the difference in objective function values ($f(x) - f(x')$) and the current temperature T. This temperature decreases with the cooling factor α after a certain number of iterations equal to the Markov chain length are performed.

SA stops and returns the best found solution x^* once the temperature T drops below a predefined stopping temperature. Figure 5.20 provides an overview of the SA procedure that we use.

```
Generate initial solution x
Set x* = x
Set T = StartTemp

While T > StopTemp
    For m = 0 to MarkovLength
        x' = random solution in N(x)
        If f(x') < f(x):
            x = x'
            If f(x') < f(x*)
                x* = x'
            End
        Elseif random number(0,1) ≤ e^((f(x)−f(x'))/T)
            x = x'
        End
    End
    T = αT
End
Result = x*
```

Figure 5.20; Simulated annealing heuristic

5.3 Solution design conclusion

In this chapter, we stated our solution design for solving the model described in Chapter 4. In short, our solution design consists of an approximate approach, where we solve the problem by means of heuristics to obtain good quality solutions in a computationally efficient manner. Section 5.1 discusses six approaches capable of constructing an initial solution where we load the workload of an incoming order in available production capacity within the periodic loading framework whilst considering the workloads of already existing orders. These six approaches, namely FL, CFL, HL, UFL, UCL and UHL are construction heuristics and apply several loading techniques from the literature in constructing initial solutions, namely forward loading and backward loading. Moreover, all six approaches immediately check whether it is more beneficial to use nonregular capacity (overtime) to prevent tardiness of either the incoming order (FL, CLF and HL) or both the incoming order and existing orders (UFL, UCL and UHL).

After we obtain an initial solution by means of the approaches from Section 5.1, we might improve with the use of the improvement heuristics that we describe in Section 5.2. These improvement heuristics are an adaptation to the steepest descent heuristics and simulated annealing.

6. Experimental design

This chapter presents the experimental design of the research, where the goal is to find the best performing solution approach from Chapter 5 in the setting of TOMRA. To realize this, we experiment on different problem instances, where an incoming order request requires to be loaded. Section 6.1 describes the experiment setting of our research, indicating how we design experiments that can provide the desired results. Additionally, Section 6.2 provides the results of different problem instances per solution approach and provides an evaluation of the results. At last, Section 6.3 concludes the chapter.

6.1 Experiment setting

As mentioned before, we aim to find the best performing solution approach from Chapter 5 in terms of objective value in our setting. This relates to the best approach to load the workload of an incoming order in available capacity, providing a due date whilst considering existing orders that have not been completed yet. Furthermore, we stress that a solution should be obtained in competent time. This is an important aspect to TOMRA, since they desire to provide order to due dates to their incoming order requests in a timely manner.

In total, we evaluate the performance of 18 solution approaches with the purpose of finding the best solution approach that TOMRA constantly can use for loading the workloads of future incoming orders:

- We first assess the performance of each of the 6 construction heuristics from Section 5.1 that are capable of loading the workload of an incoming order (FL, CFL, HL, UFL, UCL and UHL).
- Afterwards, we evaluate the performance of each of the 6 construction heuristics in combination with the two improvement heuristics from Section 5.2 to assess whether remarkable improvements occur and which improvement heuristic shows the best performance. More specifically, we evaluate the performance of the 6 construction heuristics in combination with the adapted steepest descent (ASD) improvement heuristic as well as the performance of the 6 construction heuristics combined with the simulated annealing (SA) improvement heuristic.

In our experiment setting, we try to imitate the production situation at TOMRA. This implies that we consider three production departments $S = \{1,2,3\}$ and a time horizon of 20 weeks ahead in time $T = \{0, \dots, 20\}$. In addition, we include a slack of one week ($q = 1$) and set the first two upcoming weeks as the frozen period ($|H| = 2$). We refer to Appendix C for reasoning behind further model parameters visible in Table 6.1.

Table 6.1; Model parameters values in our setting

Model parameter	Value
Exp	2
$RCE_{i'}$	10
RCT	25
ACE_1	3
ACE_2	0
ACE_3	3
ACT	50
SC	5
CO	1

Moreover, in our experiment setting, we include the identified products or operations stated Section 2.1. Appendix A provides information regarding estimations of the processing times (p_j) of these considered operations.

6.1.1 Release date determination

We consider a total of 12 raw materials with stochastic external lead times that can influence release dates of several operations. We identified this selection of materials based on recommendations and interviews with production managers and employees who are active on the work floor at TOMRA. For determining the release dates for operations that require one or multiple of these raw materials, we make use of the sample average approximation (SAA) approach mentioned in Section 4.4.

To utilize this approach, we used historical data on multiple external lead time observations for all 12 materials. Such an observation implies the date from ordering the material until the date of receiving it, which we transformed to weeks instead of days to be in line with our modeling approach. We transformed the data into weeks by rounding the number of days the delivery lasted to the nearest week. Note that there might be correlation within the historical dataset regarding external lead time observations. For instance, a supplier could have had delivering issues, resulting in multiple consecutive long external lead time observations of a certain material type. In our analysis, we ignore these types of correlations, which can have an impact on the results.

Eventually, each external lead time observation of raw material l with stochastic external lead times is a scenario ω_l. To construct a solution, we use the external lead times $L_l(\omega_l)$ of all materials l based on a 75th percentile from the historical data. We select this percentile, as we aim to seek for robust solutions and feel like this provides a reasonable balance between solution feasibility and having unnecessary late release dates. Note that choosing a different percentile can influence the results. Additionally, we assess the feasibility of the resulting solutions by means of 25 random scenarios for external lead times to obtain the service level λ. To be consistent with comparing the approaches, we use the same random scenarios for all approaches to obtain λ. Appendix F provides detail regarding the determination of release dates for operations of incoming orders in our problem setting (both deterministic r_j and stochastic $r_j(\omega_l)$).

6.1.2 Performance measures

To assess the performances of the solution approaches, we use several performance measures that can give us insight on how well a solution approach performs.

We divide the performance of each solution approach into 5 elements, namely the number of tardy orders, total overtime usage, objective function value, service level and the computational time in seconds. Both the objective function value and service level (λ) directly follow from the model described in Chapter 4. More specifically the objective function value refers to the total costs function from the model stated in Section 4.3 and the service level (λ) relates to feasibility of the resulting solution based on the SAA approach as explained above and mentioned in Section 4.4.

Next to these two measures, we also provide information regarding the number of tardy orders and the total overtime usage within the planning horizon. Both these measures are already embedded within the objective function, however mentioning these measures explicitly can provide a better understanding regarding the performance of a solution approach. At last, we also include the computational time as a measure, where the computational time (in seconds) implies the time it takes for the approach to establish the resulting solution. The reasoning behind this is that a solution within reasonable and competent computational time is desired.

6.1.3 Performance evaluation

Within our experiment setting, we evaluate the performance of each solution approach based on multiple problem instances as well as multiple unique incoming orders.

We evaluate the performance of each solution approach on a total of 4 different problem instances. These instances represent different situations with respect to the number of existing orders, production capacity and material availability. We base these instances on historical company data during past peak demand periods. We evaluate the performance of the solution approach on multiple instances, as this provides more generalized results and diminishes the chance of good performance of an approach due to coincidence. We use 4 problem instances and not more, since we desire to evaluate the performances on recent situations instead of situations of years ago with different production environments (some other operations and ways of operating). Table 6.2 presents an overview of the information per experiment instance, where we depict the number of existing orders as well as the period where an incoming order requests the due date. Appendix D provides a more detailed overview of the experiment instances.

Table 6.2; Overview of experiment instances

Instance	**1**	**2**	**3**	**4**
Existing orders	332	309	290	432
Requested due date ($cd_{i'}$)	4	6	5	4

For each of the 4 problem instances, we choose several types of incoming orders that request a due date during a busy week with few or none available regular capacity. The reason behind requesting a due date in busy week is to tackle our stated action problem from Chapter 1, since requesting a due date where the preceding weeks have enough regular capacity remaining is currently not a problem. In addition, we select multiple types of incoming orders to, again, provide more generalized results regarding the performance of the solution approaches.

In total, we choose 4 different incoming orders with the following characteristics:

- An incoming order with a relatively small workload at all departments
- An incoming order with a relatively large workload at all departments
- An incoming order with a workload at departments 1 and 2
- An incoming order with a workload at departments 2 and 3

Needless to say, evaluating performances on more different types of incoming orders would result in more robust performance results of the solution approaches. We select 4 types of incoming orders, instead of more or conducting a simulation study related to this aspect, since we feel that these types cover common characteristics of incoming orders at TOMRA and can provide us helpful insights in a timely manner. In Table 6.3, we highlight the characteristics of the 4 different incoming orders. Appendix E provides more details regarding these 4 hypothetical incoming orders.

Table 6.3; Characteristics of incoming orders

Incoming orders		**1**	**2**	**3**	**4**
Department $s = 1$	Workload (hours)	10	40	14	X
	Operations	4	10	6	X
Department $s = 2$	Workload (hours)	8.5	30	13	9
	Operations	6	15	6	5
Department $s = 3$	Workload (hours)	10	40	X	22
	Operations	1	4	X	3
Total workload (hours)		28.5	110	27	31
Total operations		11	29	12	8

6.1.4 Data preparation

As mentioned before, we use historical company data for our 4 different problem instances. This data consists of orders in the system that have not been completed yet with their corresponding operations. Moreover, the already assigned due dates of these existing orders are available. The following adjustments and additions to the historical data of existing orders are made:

- Included the processing times p_j to every operation j that we consider at TOMRA. These processing times are estimations, which can be seen in Appendix A.
- Included a release date r_j to every operation j, where we use the logic explained in Appendix F for the setting of TOMRA.
- Included the initial loading periods z_j for all operations j within the existing orders of all problem instances. We base these values on the current production setting of TOMRA (see Section 2.2), where each order is loaded one period before the assigned due dates.
- Improved on the initial loading periods z_j of the operations within existing orders for all instances by performing an extensive simulated annealing run: starting temperature of 150, stop temperature of 1, Markov chain length of 200 and a cooling factor α of 0.99. The reasoning behind this step is that the quality of the initial loading periods z_j can greatly influence the results and we, therefore, try to improve this quality first. Especially the results of the FL, CFL and HL approaches can be influenced by non-efficient z_j values of existing orders, since these approaches do not change the loading periods of these existing orders. Moreover, we improve on the z_j values, as our goal is the find a solution approach that provides the most efficient solutions in the long run with constantly high quality z_j values when incoming orders arrive.
- Included the (historical) regular and non-regular production capacity (Q_{st} and MO_{st}) per department per week during the planning horizon for every instance. In our problem setting, the capacity depends on the employee availability. Appendix D elaborates on how we obtain the exact values of Q_{st} and MO_{st} for all instances.

6.2 Experiment results

We implement the solution approaches related to the deterministic model from Section 4.3 together with the stochastic extension of Section 4.4 in Python 3.9 on a computer with AMD Ryzen 7 4700U running at 2.00 GHz with a 16 GB RAM. This section highlights the results of all solution approaches in three subsections: Section 6.2.1 discusses the results of all 6 construction heuristics, Section 6.2.2 contains the results of all 6 construction heuristic combined with adapted steepest descent (ASD) and Section 6.2.3 highlights the results of all 6 construction heuristics combined with simulated annealing (SA).

For the solution approaches consisting of an improvement heuristic (either ASD or SA), we performed 3 replications to obtain more statistically significant results. Moreover, we outline the process of determining the parameters for both improvement heuristics in Appendix G.

6.2.1 Construction heuristics results

Table 6.4 provides an overview of the results of the performance related to the 6 construction heuristics. The results depicted in the table are the average performance measures of all 4 incoming orders per problem instance.

Table 6.4; Construction heuristics results (average of the incoming orders per instance)

Instance	Approach	Objective value	Time (seconds)	Tardy orders (#)	Total Overtime (hours)	Service level (λ)
Instance 1	FL	396.9	1.7	0.8	24.7	0.60
	CFL	387.9	1.9	0.8	23.9	0.92
	HL	364.1	1.9	0.8	20.4	0.92
	UFL	1248.5	6.3	2.8	45.2	0.86
	UCL	1182.1	6.8	2.8	41.1	0.92
	UHL	489.1	3.9	2.8	24.6	0.92
Instance 2	FL	942.1	2.0	2.5	55.3	0.83
	CFL	930.7	1.9	2.5	55.5	0.87
	HL	921.9	1.7	2.3	55.9	0.87
	UFL	1703.3	1.9	3.0	89.6	0.83
	UCL	1660.8	2.3	3.0	89.3	0.83
	UHL	969.8	2.8	3.3	56.6	0.83
Instance 3	FL	523.1	1.5	1.0	40.2	0.87
	CFL	512.7	1.5	1.0	39.1	0.90
	HL	479.5	1.5	1.0	34.7	0.90
	UFL	833.4	1.7	1.0	53.0	0.94
	UCL	821.3	2.0	1.0	52.3	0.94
	UHL	466.7	2.7	1.0	33.4	0.94
Instance 4	FL	1579.3	2.8	2.8	85.6	0.61
	CFL	1568.7	2.8	2.8	84.9	0.85
	HL	1564.4	2.9	2.8	84.4	0.85
	UFL	2502.2	66.0	24.8	76.0	0.61
	UCL	2484.2	66.9	24.8	75.0	0.85
	UHL	2482.5	61.2	20.8	78.5	0.85

From the results in Table 6.4, we observe the following:

- The results change significantly with the four different problem instances. An explanation for this is that every instance has different existing orders as well as a different production capacity per department during the time horizon. Especially instance 4 has remarkably more existing orders when the hypothetical incoming orders arrive (See Section 6.1.3), resulting in either more overtime usage or more tardy orders for this instance.

- The UFL and UCL approaches show poor performance related to the objective values on all problem instances. This is mostly due to the higher overtime usage and larger number of tardy orders for all instances. Moreover, a reason for the poor performances of these approaches can be that the initial loading periods (z_j) are already very efficient before the arrival of the new order request. This can explain why unloading the operations from existing orders with due dates outside the frozen period, thus possibly changing their loading periods, results in poor performances.
- The service levels are the lowest on both forward loading approaches (FL and UFL), whilst the service levels of the remaining approaches are, for all instances, well above the 75th percentile used for constructing the solution.
- All unloading approaches (UFL, UCL and UHL) require on average more time to construct a solution as a result of their way of loading. The unloading approach require especially more time for instance 4 due to the higher number of existing orders in this instance.
- The HL approach shows the overall best objective value performances, with only the UHL approach outperforming this HL approach in the problem instance 3. A reason why UHL performs better for problem instance 3 is that the initial loading periods (z_j) are not very efficient for this instance, resulting that unloading all operations of orders with assigned due dates outside the frozen periods shows better performance.

6.2.2 Adapted steepest descent results

For the adapted steepest descent heuristic (ASD), we used a time limit of 300 seconds, a maximum iterations of 250 for the FL, CFL, HL and UHL construction heuristics and a maximum number of iterations of 50 for the UFL and UCL construction heuristics. We select these parameter settings, since these setting provide us with the most improvements in objective values in our problem setting. Appendix G describes the steps in tuning these parameters in more detail.

Table 6.5 highlights the results related to the performance of the 6 construction heuristics with the ASD improvement heuristic, where we denote the difference with the values from just the construction heuristics (Table 6.4) in brackets. The results in the table are the average performance measures of all 4 incoming orders per problem instance. For these solution approaches, we perform 3 replications per observation to obtain more generalized results.

Table 6.5; Combined construction heuristics and ASD results (average of the incoming orders per instance that are performed with 3 replications)

Instance	Approach	Objective value	Time (seconds)	Tardy orders (#)	Total Overtime (hours)	Service level (λ)
Instance 1	FL	360.4 (-36.5)	137.4 (+135.7)	1.1 (+0.3)	19.8 (-4.9)	0.70 (+0.10)
	CFL	357.8 (-30.1)	133.0 (+131.1)	1.0 (+0.2)	19.5 (-4.4)	0.88 (-0.04)
	HL	334.4 (-29.7)	149.7 (+147.8)	0.8 (+0.0)	17.4 (-3.0)	0.86 (-0.06)
	UFL	712.4 (-536.1)	292.6 (+286.3)	2.1 (-0.7)	23.6 (-21.6)	0.86 (+0.00)
	UCL	717.6 (-464.5)	286.9 (+280.1)	2.1 (-0.7)	24.4 (-16.7)	0.92 (+0.00)
	UHL	387.0 (-102.1)	131.7 (+127.8)	1.9 (-0.9)	19.5 (-5.1)	0.85 (-0.07)
Instance 2	FL	910.1 (-32.0)	168.5 (+166.5)	2.8 (+0.3)	55.0 (-0.3)	0.72 (-0.11)
	CFL	907.6 (-23.1)	107.5 (+105.6)	2.7 (+0.2)	55.1 (-0.4)	0.86 (-0.01)
	HL	893.1 (-28.8)	153.1 (+151.4)	2.6 (+0.3)	51.4 (-4.5)	0.83 (-0.04)
	UFL	1271.3 (-432.0)	175.6 (+173.7)	3.5 (+0.5)	67.4 (-22.2)	0.82 (-0.01)
	UCL	1162.0 (-498.8)	226.3 (+224.0)	3.8 (+0.8)	55.1 (-34.2)	0.85 (+0.02)
	UHL	940.1 (-29.7)	111.5 (+108.7)	4.1 (+0.8)	56.7 (+0.1)	0.85 (+0.02)
Instance 3	FL	500.4 (-22.7)	76.7 (+75.2)	1.3 (+0.3)	36.6 (-3.6)	0.88 (+0.01)
	CFL	497.1 (-15.6)	45.0 (+43.5)	1.2 (+0.2)	38.0 (-1.1)	0.90 (+0.00)
	HL	479.1 (-0.4)	34.0 (+32.5)	1.2 (+0.2)	35.1 (+0.4)	0.91 (+0.01)
	UFL	642.7 (-190.7)	222.9 (+221.2)	3.0 (+2.0)	32.2 (-20.8)	0.96 (+0.02)

	UCL	638.4 (-182.9)	192.9 (+190.9)	3.3 (+2.3)	32.9 (-19.4)	0.96 (+0.02)
	UHL	439.6 (-27.1)	192.7 (+190.0)	1.3 (+0.3)	28.7 (-4.7)	0.94 (+0.00)
Instance 4	FL	1572.9 (-6.4)	51.8 (+49.0)	3.1 (+0.3)	83.9 (-1.7)	0.61 (+0.00)
	CFL	1561.4 (-7.3)	51.4 (+48.6)	2.8 (+0.0)	84.5 (-0.4)	0.85 (+0.00)
	HL	1561.1 (-3.3)	42.3 (+39.4)	2.8 (+0.0)	84.7 (+0.3)	0.85 (+0.00)
	UFL	1998.2 (-504.0)	199.7 (+133.7)	14.9 (-9.9)	73.6 (-2.4)	0.86 (+0.25)
	UCL	2010.0 (-474.2)	192.9 (+126.0)	14.9 (-9.9)	79.0 (+4.0)	0.88 (+0.03)
	UHL	2098.1 (-384.4)	300.0 (+238.8)	14.9 (-5.9)	82.6 (+4.1)	0.85 (+0.00)

From the results in Table 6.5, we see the following:

- The improvements for most instances do not last the full 300 seconds on average. This indicates that the ASD procedure stops, as it often is not capable of finding a better neighborhood solution within the maximum number of iterations. Moreover, the times of the unloading approaches are on average longer, which can be explained by the fact that these approaches generate worse initial solutions as seen in Table 6.4.
- The service levels after applying the ASD heuristic differ from the approach of solely using construction heuristics (Table 6.4). This implies that the ASD improvement heuristics impacts the service levels. Again, especially the FL approach has low service levels in comparison with the other approaches.
- The HL approach performs best in objective value and has on average least number of tardy orders and overtime hours within the planning horizon. An explanation for this observation is that the HL approach loads the operations at some production departments in a backwards manner, considering the customer-requested due date and resulting in the least amount of tardiness for the incoming order.

6.2.3 Simulated annealing results

For the simulated annealing (SA) improvement heuristic, we used a start temperature of 30 with a stop temperature of 5, a Markov chain length of 150 and a cooling factor α of 0.975 for all 6 construction heuristics. We select these parameter settings, since, in our eyes, these settings result in the best trade-off between computational time and objective value in our problem setting, as we aim to find good quality solutions in a timely manner. Appendix G describes more detailed logic behind these parameter settings.

Table 6.6 provides an overview of the results regarding the performance of the 6 construction heuristics with the SA heuristic, with the differences compared with just the construction heuristics (Table 6.4) in brackets. Once again, the results in the table are the average performance measures of all 4 incoming order per instance, where we perform 3 replications per observation.

Table 6.6; Combined construction heuristics and SA results (average of the incoming orders per instance that are performed with 3 replications)

Instance	Approach	Objective value	Time (seconds)	Tardy orders (#)	Total Overtime (hours)	Service level (λ)
Instance 1	FL	338.9 (-58.0)	1304.4 (+1302.7)	1.1 (+0.3)	19.5 (-5.2)	0.72 (+0.12)
	CFL	346.9 (-41.0)	1301.2 (+1299.3)	1.3 (+0.5)	20.2 (-3.7)	0.93 (+0.01)
	HL	335.9 (-28.2)	1300.1 (+1298.2)	0.9 (+0.1)	18.9 (-1.5)	0.92 (+0.00)
	UFL	547.3 (-701.2)	1385.4 (+1379.1)	1.6 (-1.2)	22.5 (-22.7)	0.78 (-0.08)
	UCL	573.2 (-608.9)	1378.6 (+1371.8)	2.1 (-0.7)	21.7 (-19.4)	0.87 (-0.05)
	UHL	390.3 (-98.8)	1382.3 (+1378.4)	2.0 (-0.8)	17.9 (-6.7)	0.92 (+0.00)
Instance 2	FL	901.5 (-40.6)	1124.3 (+1122.3)	2.3 (-0.2)	51.9 (-3.4)	0.69 (-0.14)
	CFL	900.3 (-30.4)	1132.1 (+1130.2)	2.8 (+0.3)	55.8 (+0.3)	0.85 (-0.02)
	HL	891.2 (-30.7)	1123.0 (+1121.3)	2.6 (+0.3)	50.4 (-5.5)	0.82 (-0.05)
	UFL	1076.1 (-627.2)	1130.4 (+1128.5)	4.0 (+1.0)	55.5 (-34.1)	0.74 (-0.09)
	UCL	1048.6 (-612.2)	1133.7 (+1131.4)	4.3 (+1.3)	53.5 (-35.8)	0.89 (+0.06)

	UHL	912.5 (-57.3)	1132.6 (+1129.8)	3.6 (+0.3)	57.6 (+1.0)	0.88 (+0.05)
Instance 3	FL	498.1 (-25.0)	1197.2 (+1195.7)	1.4 (+0.4)	37.9 (-2.3)	0.93 (+0.06)
	CFL	495.0 (-17.7)	1214.8 (+1213.3)	1.3 (+0.3)	36.2 (-2.9)	0.94 (+0.04)
	HL	477.4 (-2.1)	1205.6 (+1204.1)	1.2 (+0.2)	37.2 (+2.5)	0.94 (+0.04)
	UFL	572.3 (-261.1)	1196.1 (+1194.4)	2.2 (+1.2)	33.3 (-19.7)	0.81 (-0.13)
	UCL	558.7 (-262.6)	1197.3 (+1195.3)	2.2 (+1.2)	33.0 (-19.3)	0.92 (-0.02)
	UHL	440.7 (-26.0)	1199.7 (+1197.0)	1.3 (+0.3)	28.3 (-5.1)	0.94 (+0.00)
Instance 4	FL	1573.0 (-6.3)	1308.3 (+1305.5)	2.8 (+0.0)	85.5 (-0.1)	0.61 (+0.00)
	CFL	1562.8 (-5.9)	1309.2 (+1306.4)	2.9 (+0.1)	84.2 (-0.7)	0.85 (+0.00)
	HL	1563.5 (-0.9)	1306.8 (+1303.9)	2.8 (+0.0)	86.1 (+1.7)	0.88 (+0.03)
	UFL	2083.0 (-419.2)	1356.7 (+1290.7)	15.2 (-9.6)	80.4 (+4.4)	0.61 (+0.00)
	UCL	2058.8 (-425.4)	1355.1 (+1288.2)	14.8 (-10.0)	79.1 (+4.1)	0.81 (-0.04)
	UHL	2015.1 (-467.4)	1347.6 (+1286.4)	12.3 (-8.5)	78.6 (+0.1)	0.79 (-0.06)

From the results in Table 6.6, we observe the following:

- The solution approaches with the SA improvement heuristic show similar performance in terms of objective value compared to the approaches including the ASD heuristic. However, the approaches with the SA heuristic require significantly more computation time to release an outcome.
- Both the FL and UFL have low service levels
- Apart from instance 3, the HL approach shows the best performance in terms of objective values.

6.2.4 Summarized results

Table 6.7 summarizes the results of all solution approaches. The performance measures depicted in the table are the average values of all instances of Table 6.4, 6.5 and 6.6.

Table 6.7; Average results of all approaches

Approach		Objective value	Time (seconds)	Tardy orders (#)	Total Overtime (hours)	Service level (λ)
Construction heuristic	FL	860.3	2.0	1.8	51.5	0.73
	CFL	850.0	2.0	1.8	50.9	0.89
	HL	832.5	2.0	1.7	48.9	0.89
	UFL	1571.8	19.0	7.9	65.9	0.81
	UCL	1537.1	19.5	7.9	64.4	0.89
	UHL	1102.0	17.6	6.9	48.3	0.89
Construction heuristic + Adapted steepest descent (ASD)	FL	835.9	108.6	2.1	48.8	0.73
	CFL	831.0	84.2	1.9	49.3	0.87
	HL	816.9	94.8	1.9	47.2	0.86
	UFL	1156.1	222.7	5.9	49.2	0.88
	UCL	1132.0	224.7	6.0	47.9	0.90
	UHL	966.2	188.2	5.5	46.9	0.87
Construction heuristic + Simulated annealing (SA)	FL	827.9	1233.5	1.9	48.7	0.74
	CFL	826.3	1239.3	2.1	49.1	0.89
	HL	817.0	1233.9	1.9	48.1	0.89
	UFL	1069.7	1267.2	5.7	47.9	0.74
	UCL	1059.8	1266.2	5.8	46.8	0.87
	UHL	939.6	1265.6	4.8	45.6	0.88

From the summarized results, we see that the approaches with only the construction heuristic seem to perform well on average. However, both improvement heuristics manage to improve on the objective values of the initial solutions. Figure 6.1 shows the average improvement in the objective values for both the ASD and SA improvement heuristics. From this figure, we clearly see that the SA heuristic results in more improvement for most construction heuristics. Moreover, both heuristics show more improvement in the unloading approaches that, on average, have worse initial objective values. Nevertheless, all unloading approaches still do not outperform the approaches that do not consider reloading operations from existing orders with already assigned due dates after applying the improvement heuristics with the current parameter settings.

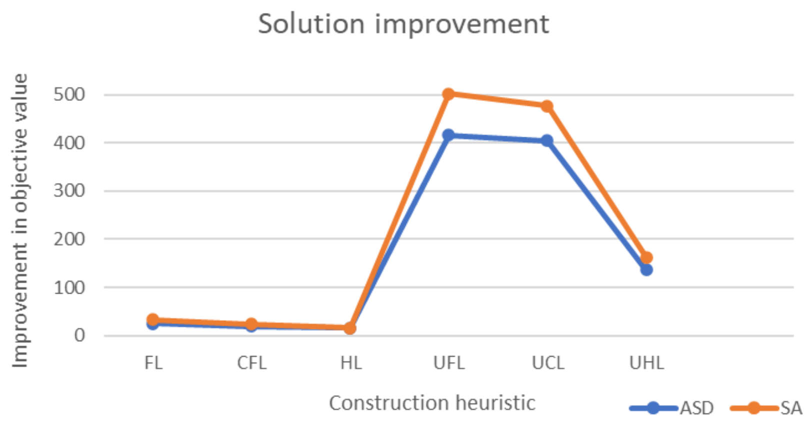

Figure 6.1; Average solution improvements per construction heuristic for the adapted steepest descent (ASD) and simulated annealing (SA) improvement heuristics

Moreover, we observe that the HL construction heuristic shows the overall best performances in terms of objective value for just the construction heuristic itself as well as in combination with both improvement heuristics. Additionally, the HL approach shows solid service levels in comparison with the other approaches. The average initial solution value for the HL approach of 832.5 is already good performance in comparison with other approaches. However, the HL approach in combination with the ASD heuristic with a value of 816.9 shows notable improvement in the objective value. The average performance of the HL approach in combination with the SA heuristic is similar, namely 817.0, however this is achieved in much more computational time on average (1233.9 seconds of SA in comparison to 94.8 seconds of ASD).

Additionally, we see that the unloading approaches (UFL, UCL and UHL) do show higher objective values than the approaches that do not consider reloading operations from existing orders with already assigned due dates. This poor performance can be explained by the fact that these approaches first unload all orders with an assigned due date outside the frozen periods and treat the incoming order like an existing order. As our parameter settings assign more cost to tardiness of existing orders in comparison to tardiness of incoming orders, these higher objective values can be explained.

Furthermore, we see that the forward loading approaches have the lowest service levels in general. The low service levels for the forward loading approaches indicate that these approaches try to load the operations that require materials with stochastic external lead times as soon as their release date, causing these operations to be loaded in periods that precede their sampled release date. These low service levels are a direct result of the 75th percentile we use for determining the release dates of the operations requiring one or multiple materials with stochastic external lead times. A higher percentile would most likely result in higher service levels for these forward loading approaches. Note that this service level is also based on 25 random scenarios, however the service levels of the forward loading approaches will remain the lowest with other scenarios, as the other approaches use the same random scenarios for evaluating the service level.

6.3 Experimental design conclusion

In this chapter, we conducted experiments on the solution approaches from Chapter 5 with the aim of providing good quality solutions in a timely manner for the model of Chapter 4. We developed an experimental design for comparing 18 solution approaches: 6 construction heuristics, 6 construction heuristics combined with ASD and 6 construction heuristics combined with SA.

Section 6.1 describes the experiment setting for comparing these solution approaches. We evaluate the performance of these approaches based on their objective values, computational times, number of tardy orders, total overtime usages and service levels. The service levels relate to loading the operations that require one or multiple materials with stochastic external lead times. For determining the release dates of these operations, we use the SAA approach from Section 4.4. In our setting, we apply a 75^{th} percentile for determining the release dates and assess the feasibility of the loaded operations based on 25 random scenarios. Eventually, we evaluate the performances of the approaches on 4 problem instances that are based on historical order data with 4 different incoming orders at each instance that all request a due date during a peak demand period.

Section 6.2 highlights the results of the performed experiments, where the HL construction heuristics shows the overall best performance in terms of objective value. Moreover, the forward loading approaches (FL and UFL) tend to show low service levels as a result of their way of loading operations requiring one or multiple materials with stochastic external lead times. At last, both improvement heuristics (ASD and SA) improve on the objective value, where the ASD heuristic converges faster to similar solution values compared to the SA heuristic.

7. Conclusion and discussion

This chapter presents the conclusions and recommendations of our research. More specifically, Section 7.1 recapitulates the main findings. Section 7.2 provides recommendations to TOMRA. Section 7.3 discusses the limitations of our research as well as future research areas. Finally, Section 7.4 addresses the academic relevance of this research.

7.1 Conclusion

The production facility of TOMRA in Apeldoorn often experiences production related problems during peak demand periods of the backroom elements of bottle deposit systems. Among other things, high workloads and inventory shortages make up for these production problems. These problems are mainly caused by the fact that the production capacity is currently not fully considered in production planning activities at TOMRA. More specifically, optimizing due dates assignments to incoming orders is of great relevance, where customers currently determine the due dates. As TOMRA does not have detailed capacity overviews, they have little saying in providing feedback on the requested due dates and almost always comply with the customers' requests resulting in heavy workload fluctuations. Therefore, the main research question is the following:

"How to optimize the combined process of assigning due dates with spreading the workload of incoming orders?

We developed a finite loading model that is capable of optimizing due date assignments to incoming order requests in a setting with multiple production departments, where customers have a desired due date and no incoming orders are refused. The model considers imposing tardiness to existing orders that have not been completed yet to fit an incoming order in the production capacity. Moreover, we consider the usage of nonregular capacity in this process. In addition, we developed a sample average approximation (SAA) inspired approach that is capable of determining release dates of operations that require one or multiple materials with stochastic external lead times.

The solution approach for solving the model in our setting consists of an approximate method to obtain good quality model solution in a timely manner by means of heuristics. In total, we considered 6 construction heuristics that can generate initial solutions to our problem, namely FL, CFL, HL, UFL, UCL and UHL. Moreover, we included two improvement heuristics that can improve on initial solutions, namely ASD and SA.

To find the best solution approach in our problem setting at TOMRA, we developed an experimental design that evaluates the performances of the approaches based on 4 problem instances. Each problem instance requires loading of 4 different incoming orders that all request a due date during a peak demand period.

From the experiment results, we conclude that the HL approach, which loads operations forward or backward depending on preferences of customers and production departments, shows the overall best performance in terms of objective value. Moreover, both forward loading approaches (FL and UFL) show poor performance related to the service levels as a result of their way of loading the operations. Using only the HL construction heuristic provides already good quality solution. However, applying an improvement heuristic to change some loading periods of order to fit the workload of an incoming order even better within available production capacity improves the objective values. ASD shows similar performance related to the SA improvement heuristic, however does so in less computational time, making this approach superior with its current parameters in our problem setting.

7.2 Recommendations

Based on the evaluated solution approaches, we recommend TOMRA to implement the HL approach for loading the workloads of incoming orders. This construction heuristic outperforms the other solution approaches that we considered during this research. Moreover, we recommend applying the ASD improvement heuristic to enhance the quality of solutions in a timely manner.

For the implementation of this approach, several changes are required at TOMRA. The first of which being to make use of the estimated processing times during this research to gain detailed insights on the workloads of orders. Additionally, the employee availability should be monitored on a weekly basis to obtain the production capacity, where we recommend establishing this per production department as far ahead in time as possible. At last, a user-friendly dashboard would help in filling in the order elements of incoming orders requests as well as their requested due date after which the recommended best solution approach of this research can load the operations per department.

Furthermore, we recommend generating capacity overviews per production department. These overviews can contain the workload in hours per week for the planning horizon at a certain production department. Since TOMRA currently has no capacity related overviews, this would help in showcasing how much workload is planned in the upcoming weeks and to gain insights into their regular and nonregular capacity usage.

Finally, we stress the importance of customers having the exact specifications for the table and conveyor departments available well ahead of the due date. The MTO elements of Dutch orders are dependent on checks at client locations to begin their production. As of right now, these checks are often executed only a couple of days before the due date. This causes that TOMRA is not able to begin the production activities of the tables and conveyors for these orders well ahead of time, resulting in less possibilities to spread workloads and less efficient production schedules.

7.3 Limitations and future research

We limited the scope of our research to the most backroom elements that TOMRA produces, however some less common products or services that TOMRA delivers were excluded. Moreover, we did not consider all materials that TOMRA uses for producing the backroom elements for determining (stochastic) release dates.

In addition, we did not implement the solution design of the model with the use of the fastest programming language. This limited the number of neighborhood solutions that we could construct in a certain time limit. For instance, this causes the ASD improvement heuristic to not go through a larger proportion of the neighborhood in the search for better neighborhood solutions. Besides, the speed of our programming language resulted in not further tuning the parameters for both improvement heuristics.

An area for future research is to develop more loading techniques capable of constructing initial solutions. In our research, we only considered 6 construction heuristic. There can be other approaches that can potentially outperform our best found solution approach. Moreover, we implemented a specific way to immediately see if it is worth to use some extra nonregular capacity to prevent tardiness. This procedure can be extended or adjusted in various ways to improve the results.

In the future, more extensive neighborhood operators can be applied for reloading operations in different manners to make the improvement heuristics generate better solutions. For example, a bottleneck operation can swap with an operation that is loaded in an earlier week, which can possibly also be swapped with an operation loaded in an earlier week etcetera, to consider a wide range of weeks in constructing a neighborhood solution.

Another area for future research is to include stochasticity related to the production capacity. The capacity at TOMRA depends on the employee availability, of which the exact availability is hard to know multiple weeks or months ahead in time. The employee availability becomes more uncertain further in the future, which can influence the loaded workloads during these periods. One could consider the production capacity as a stochastic element to make the model more realistic. Moreover, some employees can work at multiple production department at TOMRA. The model could be changed in such a way to arrange the optimal distribution of the employees over the production departments depending on existing and incoming workloads. Besides, some employees with multiple years of experience can produce some types of backroom elements faster than others. This can influence the processing times, resulting in deviated capacity insights.

At last, an extension to the model can be to make distinctions between different customers. This way, TOMRA can prioritize between customers, which is something they are planning on doing in the near future.

7.4 Academic relevance

The finite loading method that we propose in this research can be applied to a wide range of production environments. Both the ideas that nonregular capacity can be used to fit the workloads of orders and that the workloads of existing orders can be reloaded to other periods are extremely relevant in our eyes. The deterministic model from Section 4.3 that we use during this research requires a customer-requested due date an input parameter, but can easily be adapted if this is not the case within another setting. Additionally, production environments might require precedence relationships between production departments for which an adaptation to the current model is required via an extra constraint.

Furthermore, the stochastic element that we include within our model formulation that arranges the determination of release dates based on the concept of SAA can be relevant for various problem settings. Scheduling problems that require release dates as input can make use of the method that we propose in Section 4.4 if there are materials with stochastic external lead times.

At last, the concepts behind the construction heuristics from Section 5.1 that are capable of finding good quality and computationally efficient solutions can be applied in other problem settings as well. Whilst the concepts of forward and backward loading already are well-known within the literature, the ideas of collective loading and loading the orders differently depending on customer preferences or the production department (hybrid loading) add academical value to our best knowledge.

References

Aickelin, U., & Clark, A. (2011). Heuristic optimisation. Journal of the Operational Research Society, 62(2), 251–252. doi:10.1057/jors.2010.160

Bertrand, J. W. M. (1983). The use of workload information to control job lateness in controlled and uncontrolled release production systems. Journal of Operations Management, 3(2), 79–92. doi:10.1016/0272-6963(83)90009-8

de Boer, R. (1998). Resource-constrained multi-project management: a hierarchical decision support system. Universiteit Twente.

Corti, D., Pozzetti, A., & Zorzini, M. (2006). A capacity-driven approach to establish reliable due dates in a MTO environment. International Journal of Production Economics, 104(2), 536–554. doi:10.1016/j.ijpe.2005.03.003

Gademann, N., & Schutten, M. (2005). Linear-programming-based heuristics for project capacity planning. IIE Transactions, 37(2), 153–165. doi:10.1080/07408170590885611

Glover, F., (2003). Handbook of Metaheuristics. Springer.

Glover, F., Laguna, M. (1993). Tabu search. In: Reeves, C. (ed.) Modern Heuristic Techniques for Combinatorial Problems, pp. 71–140. Blackwell Scientific Publishing, New York

Gordon, V. S., Proth, J.-M., & Chu, C. (2002). Due date assignment and scheduling: Slk, TWK and other due date assignment models. Production Planning & Control, 13(2), 117–132. doi:10.1080/09537280110069621

Gordon, V., Strusevich, V., & Dolgui, A. (2011). Scheduling with due date assignment under special conditions on job processing. Journal of Scheduling, 15(4), 447–456. doi:10.1007/s10951-011-0240-2

Goren, S., & Sabuncuoglu, I. (2009). Optimization of schedule robustness and stability under random machine breakdowns and processing time variability. IIE Transactions, 42(3), 203–220. doi:10.1080/07408170903171035

Guhlich, H., Fleischmann, M., & Stolletz, R. (2015). Revenue management approach to due date quoting and scheduling in an assemble-to-order production system. OR Spectrum, 37(4), 951–982. https://doi.org/10.1007/s00291-015-0401-3

Guo, P., Cheng, W., & Wang, Y. (2014). A general variable neighborhood search for single-machine total tardiness scheduling problem with step-deteriorating jobs. Journal of Industrial &Amp; Management Optimization, 10(4), 1071–1090. https://doi.org/10.3934/jimo.2014.10.1071

Hans, E. W. (2001). Resource Loading by Branch-and-Price Techniques. Twente University Press (TUP).

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

Heerkens, H., & van Winden, A. (2017). Solving Managerial Problems Systematically (Routledge-Noordhoff International Editions) (1st ed.). Groningen/Houten, Netherlands: Routledge.

Henderson, D., Jacobson, S. H., & Johnson, A. W. (2003). The Theory and Practice of Simulated Annealing. International Series in Operations Research & Management Science, 287–319. doi:10.1007/0-306-48056-5_10

Jing, S. (2013). Logistics scheduling with due date assignment and batch delivery. 2013 International Conference on Management Science and Engineering 20th Annual Conference Proceedings. doi:10.1109/icmse.2013.6586328

Kate, H. A. Ten. (1994). Towards a better understanding of order acceptance. International Journal of Production Economics, 37(1), 139–152. doi:10.1016/0925-5273(94)90014-0

Krim, H., Benmansour, R., Duvivier, D., & Artiba, A. (2019). A variable neighborhood search algorithm for solving the single machine scheduling problem with periodic maintenance. RAIRO - Operations Research, 53(1), 289–302. doi:10.1051/ro/2018055

Ng, C. T. D., Cheng, T. C. E., Kovalyov, M. Y., & Lam, S. S. (2003). Single machine scheduling with a variable common due date and resource-dependent processing times. Computers & Operations Research, 30(8), 1173–1185. doi:10.1016/s0305-0548(02)00066-7

Nyhuis, F., & Pereira Filho, N. A. (2002). Methods and tools for dynamic capacity planning and control. Gestão & Produção, 9(3), 245–260.

Olhager, J. (2003). Strategic positioning of the order penetration point. International Journal of Production Economics, 85(3), 319–329. doi:10.1016/s0925-5273(03)00119-1

Öztürk, A., Kayalıgil, S., & Özdemirel, N. E. (2006). Manufacturing lead time estimation using data mining. European Journal of Operational Research, 173(2), 683–700. doi:10.1016/j.ejor.2005.03.015

Pagnoncelli, B. K., Ahmed, S., & Shapiro, A. (2009). Sample Average Approximation Method for Chance Constrained Programming: Theory and Applications. Journal of Optimization Theory and Applications, 142(2), 399–416. doi:10.1007/s10957-009-9523-6

Park, C., Song, J., Kim, J.-G., Kim, I., 1999. Delivery date decision support system for the large scale make-to-order manufacturing companies. Production Planning & Control 10 (10), 585–597. https://doi.org/10.1080/095372899232885

Pirlot, M. (1996). General local search methods. European Journal of Operational Research, 92(3), 493–511. doi:10.1016/0377-2217(96)00007-0

Radar, D. J. (2010). Deterministic operations research: Models and Methods in Linear Optimization. John Wiley & Sons, Inc.

Robinson, K. R., & Moses, S. A. (2006). Effect of granularity of resource availability on the accuracy of due date assignment. International Journal of Production Research, 44(24), 5391–5414. doi:10.1080/00207540600665810

Ruiz, R. (2015). Scheduling Heuristics. Handbook of Heuristics, 1–24. doi:10.1007/978-3-319-07153-4_44-1

Saniuk, A., & Waszkowski, R. (2016). Make-to-order manufacturing - new approach to management of manufacturing processes. IOP Conference Series: Materials Science and Engineering, 145, 022005. doi:10.1088/1757-899x/145/2/022005

Sha, D., & Liu, C. H. (2004). Using Data Mining for Due Date Assignment in a Dynamic Job Shop Environment. The International Journal of Advanced Manufacturing Technology, 25(11–12), 1164–1174. https://doi.org/10.1007/s00170-003-1937-y

Shabtay, D., & Steiner, G. (2007). A survey of scheduling with controllable processing times. Discrete Applied Mathematics, 155(13), 1643–1666. doi:10.1016/j.dam.2007.02.003

Shapiro, A. (2003). Monte Carlo Sampling Methods. Stochastic Programming, 353–425. doi:10.1016/s0927-0507(03)10006-0

Shen, J., & Zhu, Y. (2018). A parallel-machine scheduling problem with periodic maintenance under uncertainty. Journal of Ambient Intelligence and Humanized Computing. doi:10.1007/s12652-018-1032-8

Silver, E. A., Pyke, D. F., & Thomas, D. J. (2021). Inventory and Production Management in Supply Chains (4th ed.). CRC Press.

Song, D. P., Hicks, C., & Earl, C. F. (2002). Product due date assignment for complex assemblies. International Journal of Production Economics, 76(3), 243–256. doi:10.1016/s0925-5273(01)00170-0

Sörensen, K., Sevaux, M., & Glover, F. (2018). A history of metaheuristics. In Handbook of heuristics (pp. 791-808). Springer, Cham.

Tahasozgen. (2021, November 5). Mathematics Stack Exchange.
https://math.stackexchange.com/questions/4297323/number-of-global-optima-in-single-solution-metaheuristics

Thürer, M., Stevenson, M., Silva, C., & Land, M. (2013). Towards an Integrated Workload Control (WLC) Concept: The Performance of Due Date Setting Rules in Job Shops with Contingent Orders. International Journal of Production Research, 51(15), 4502–4516. doi:10.1080/00207543.2013.774485

Thürer, M., & Stevenson, M. (2019). The use of finite loading to guide short-term capacity adjustments in make-to-order job shops: an assessment by simulation. International Journal of Production Research, 58(12), 3554–3569. https://doi.org/10.1080/00207543.2019.1630771

Wullink, G., Gademann, A. J. R. M., Hans, E. W., & van Harten, A. (2004). Scenario-based approach for flexible resource loading under uncertainty. International Journal of Production Research, 42(24), 5079–5098. doi:10.1080/002075410001733887

Yamashiro, H., & Nonaka, H. (2021). Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem. Operations Research Perspectives, 8, 100196. doi:10.1016/j.orp.2021.100196

Yin, Y., Cheng, T. C. E., Wu, C.-C., & Cheng, S.-R. (2013). Single-machine common due-date scheduling with batch delivery costs and resource-dependent processing times. International Journal of Production Research, 51(17), 5083–5099. doi:10.1080/00207543.2013.793424

Zorzini, M., Corti, D., & Pozzetti, A. (2008). Due date (DD) quotation and capacity planning in make-to-order companies: Results from an empirical analysis. International Journal of Production Economics, 112(2), 919–933. doi:10.1016/j.ijpe.2007.08.005

Appendix A: Processing times estimation

This appendix outlines the methods that we applied for estimating the process times of the considered products per production department. Moreover, we present the results of the estimation methods.

Table department

The table department produces tables that are customizable and can differ per customer. Yamashiro and Nonaka (2021) propose to use regression for the estimation processing times for these types of products. We apply linear regression on the characteristics of the ordered tables to estimate their processing times. We include the following characteristics:

- Length of the table expressed in meters.
- Type of table, either EasyPac (EP) or MultiPac (MUP2)
- Shape of the table, either straight or L-shaped

We used data of the table productions from the years 2022 and 2023, containing the start / end time and date of a produced table with the corresponding table characteristics. Note that, as the table characteristics, we only consider the length of tables and not the width. The reasons for this that MultiPac tables are automatically wider than EasyPac tables, the width of tables is rarely mentioned within the dataset and the width per type of table do not differ much. For instance, the width of the EasyPac tables is in somewhat standardized lengths of 0.8 and 1 meter.

After we identified the characteristics and obtained the data, we take the following actions to clean the data: 1) Remove tables with different start and end dates. 2) Remove tables that started at the exact same time. This is done, as employees sometimes select multiple tables at once to begin their production and mark them all as complete once they are all finished, resulting in combined processing times of (possibly different) tables. 3) Remove tables with unrealistic short processing times. 4) Remove outliers with processing times outside 2 standard deviations of tables with the same length.

Table A.1 provides an overview of the results following from the linear regression model, where duration indicates the process time.

Table A.1; Linear regression results of process times table department

	Estimate	*Std. Error*	*t value*	*P(>|t|)*
Intercept	2.24305	0.07841	28.606	< 2 e-16
Length	0.15981	0.02610	6.124	1.25 e-09
Type	-0.51496	0.05686	-9.057	< 2 e-16
Shape	0.30977	0.06772	4.574	5.29 e-06

From the results, all three characteristics are statistically significant, since they have a small p-value. This implies that the following equation estimates the duration of a table production:

$$Duration = 2.24305 + 0.15981 * \text{Length} - 0.51496 * \text{Type} + 0.30977 * \text{Shape}$$

$$Type = \begin{cases} 1, & if\ table\ is\ EasyPac\ (EP) \\ 0, & if\ table\ is\ MultiPac\ (MUP2) \end{cases}$$

$$Shape = \begin{cases} 1, & if\ table\ is\ L-shaped \\ 0, & if\ table\ is\ straight \end{cases}$$

To provide an example of the results, Figure A.1 shows an overview of the different table lengths (expressed in meters) within the cleaned data with their corresponding process time (expressed in hours). The line indicating the relation between length and duration is a result from the least-squares method that the linear regression model uses.

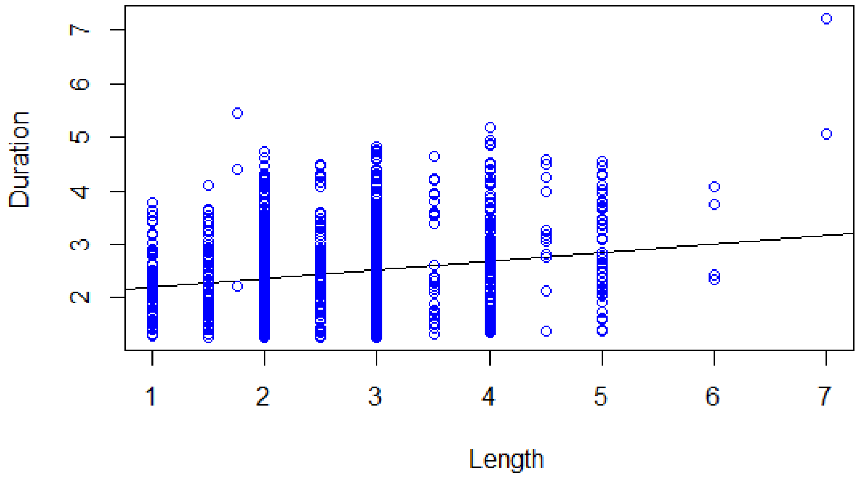

Figure A.1; Relation between length of table and process time

For existing orders, the results of the regression can be applied to estimate the process time of tables. For tables of incoming orders that are not yet within the system with detailed information about their shapes or sizes, we apply classification on the lengths. More specifically, we classify the lengths of tables by means of the following classes: Small (< 3 meter), Medium (≥ 3 & < 5 meter), Large (≥ 5 & < 10 meter) and Extra Large (≥ 10 meter). With the use of this classification, TOMRA still can make distinctions between different sizes of tables within an incoming order, that have significant different processing times, without requiring knowledge for the exact table sizes. Table A.2 provides the classification of EasyPac (EP) and MultiPac (MUP2) tables that are part of an incoming order, following from the regression results:

Table A.2; Classification tables within incoming order

Table type	**Small (< 3 meter)**	**Medium (≥ 3 & < 5 meter)**	**Large (≥ 5 & < 10 meter)**	**Extra Large (≥ 10 meter)**
Straight EP	2 hours	$2\frac{1}{3}$ hours	$2\frac{5}{6}$ hours	$3\frac{1}{2}$ hours
Straight MUP2	$2\frac{1}{2}$ hours	$2\frac{5}{6}$ hours	$3\frac{1}{3}$ hours	4 hours
Angled EP	$2\frac{1}{3}$ hours	$2\frac{2}{3}$ hours	$3\frac{1}{6}$ hours	$3\frac{5}{6}$ hours
Angled MUP2	$2\frac{5}{6}$ hours	$3\frac{1}{6}$ hours	$3\frac{2}{3}$ hours	$4\frac{1}{3}$ hours

Roller belt department

The roller belt department at TOMRA produces standardized items that do not differ per customer. TOMRA produces these items to stock them. For the products that require production at the roller belt department, no clean data is available. This implies that from the available data, we cannot derive realistic processing times for every product. Because of this, we estimate some the processing times by means of measurements. As the products in question are standardized products, we take the average of the measures to find an estimated processing time for every product. Moreover, we round these processing times up to the nearest 5 minutes to cover not too much detail in our estimations.

Additionally, some other products at the roller belt department are rarely ordered. For these items we asked several employees on their opinion on the processing times of these products to obtain an estimate.

At last, we do not consider the roller belt variants mentioned in Section 2.1.2 for occupying capacity, and therefore we do not require an estimation of their processing times. The main reason for this is that part these products are currently produced separately and the other part are outsourced to another production facility. Table A.3 provides an overview of the processing times estimation of the product variants from the roller belt department.

Table A.3; Estimated processing times for products at roller belt department

Product type	Size	Production time (hours)
Corners	All regular sizes	$\frac{1}{12}$
	All German variants	$\frac{1}{6}$
Elevator belt	0.5m	$\frac{1}{3}$
	1m	$\frac{1}{3}$
	1.5m	$\frac{5}{12}$
	2m	$\frac{1}{2}$
Elevator belt start/stop	0.5m	$\frac{5}{6}$
	1m	1
	1.5m	$1\frac{1}{6}$
	2m	$1\frac{2}{3}$
PET elevator belt	160mm	8
	470mm	3
Merger	Single size	1
V2 Conveyor	Single size	$\frac{3}{4}$
ECO-WALL	T-9/T-8 600	5
	600 E+L+B	3
	T-9 300	2

Conveyor department

The conveyor department produces products that can differ per customer in terms of sizes. For estimating the processing times of the products that require production at the conveyor department, we use both available data and expert opinions. The data consist of the start / end time and date of a produced conveyor type with the corresponding sizes.

In addition, just like with some products at the roller belt department, TOMRA obtains order requests for some types of conveyors only a few times per year. Because of this, we based the estimated processing times of these conveyors on opinions of the employees instead of incomplete or unavailable data.

Table A.4 provides an overview of the estimated processing times of the products at the conveyor department.

Table A.4; Estimated processing times for products at conveyor department

Conveyor type	**Production time (hours)**
Z-Shape	$6.4 + 0.4 * size(m)$
Feeder	12
Pre-Feeder	$4.4 + 0.4 * size(m)$
Depot	$6.4 + 0.4 * size(m)$
Wash & Waste	10

Just like with the table department, we can derive detailed processing times of the conveyors (Z-shape, Pre-Feeder and Depot) that are part of existing orders. However, for conveyors of incoming orders that are not yet within the system with detailed information about their lengths, we again apply classification of the lengths. Table A.5 outlines the classification of Z-shape conveyors, Pre-feeders and depot conveyors with the corresponding processing times:

Table A.5; Classification products conveyor department within incoming order

Conveyor type	**Small (< 5 meter)**	**Medium (≥ 5 & < 10 meter)**	**Large (≥ 10 meter)**
Z-shape	8 hours	10 hours	12 hours
Pre-feeder	6 hours	8 hours	10 hours
Depot	8 hours	10 hours	12 hours

Appendix B: Logic behind solution approaches

This appendix contains the logic behind the solution approaches from Section 5.1 to load the incoming order. In total, we provide seven flowcharts, the six approaches (FL, CFL, UL, UFL, UCL and UHL) as well as the logic behind choices of using extra overtime to prevent tardiness of orders.

FL approach

1) $x_j = z_j$ for all operations within existing orders ($i \neq i'$)
2) For all departments $s \in S$:
3) Determine total existing workloads per period $t \in T$ (WL_{st})
4) For all operations within $J_{i'}$ that require production at s ($y_j = s$):
   - 4.1) Select operation with earliest release date
     - 4.1.1) Operations with equal release dates?
       - Select operation with shortest processing time
     - 4.1.2) Operations with equal release dates and equal processing times?
       - Select first operation within $J_{i'}$
   - 4.2) Week $= r_j$
   - 4.3) No regular capacity available in Week? ($WL_{st} > Q_{st}$, where $t =$ Week)
     - If yes: Week $=$ Week $+ 1$ (Go to 4.3)
   - 4.4) Maximum overtime exceeded? ($WL_{st} + p_j > Q_{st} + MO_{st}$, where $t =$ Week)
     - If yes: Week $=$ Week $+ 1$ (Go to 4.3)
     - Otherwise: $x_j =$ Week (Go to 4.5)
   - 4.5) Update total workloads ($WL_{st} = WL_{st} + p_j$)
5) Decisions extra nonregular capacity to avoid tardiness

Figure B.1; FL approach procedure

CFL approach

1) $x_j = z_j$ for all operations within existing orders ($i \neq i'$)
2) For all departments $s \in S$:
3) Determine total existing workloads per period $t \in T$ (WL_{st})
4) For all operations within $J_{i'}$ that require production at s ($y_j = s$):
   - 4.1) Determine collective loading period ($\max_{j \in J_{i'} | y_j = s} \{r_j\}$)
   - 4.2) Select operation with earliest release date
     - 4.2.1) Operations with equal release dates?
       - Select operation with shortest processing time
     - 4.2.2) Operations with equal release dates and equal processing times?
       - Select first operation within $J_{i'}$
   - 4.3) Week $=$ Collective loading period
   - 4.4) No regular capacity available in Week? ($WL_{st} > Q_{st}$, where $t =$ Week)
     - If yes: Week $=$ Week $+ 1$ (Go to 4.4)
   - 4.5) Maximum overtime exceeded? ($WL_{st} + p_j > Q_{st} + MO_{st}$, where $t =$ Week)
     - If yes: Week $=$ Week $+ 1$ (Go to 4.4)
     - Otherwise: $x_j =$ Week (Go to 4.6)
   - 4.6) Update total workloads ($WL_{st} = WL_{st} + p_j$)
5) Decisions extra nonregular capacity to avoid tardiness

Figure B.2; CFL approach procedure

HL approach

1) $x_j = z_j$ for all operations within existing orders ($i \neq i'$)
2) Incoming order requests compliance with due date? ($RCE_{i'} > 0$)
   If yes: Go to 3
   Otherwise: Go to 8

3) For all departments $s \in S$:
4) Determine total existing workloads per period $t \in T$ (WL_{st})
5) Does department desires early completion? ($ACE_s > 0$)
   If yes: Go to 6
   Otherwise: Go to 7

6) For all operations within $J_{i'}$ that require production at s ($y_j = s$):
   6.1) Select operation with latest release date
      6.1.1) Operations with equal release dates?
         Select operation with shortest processing time
      6.1.2) Operations with equal release dates and equal processing times?
         Select first operation within $J_{i'}$
   6.2) Counter = 0
   6.3) Week = $cd_{i'} - q$
   6.4) Counter = Counter + 1
   6.5) $r_j >$ Week?
      If yes: Week = $cd_{i'} - q$ + Counter (Go to 6.4)
      Otherwise: Go to 6.6
   6.6) No regular capacity available in Week? ($WL_{st} > Q_{st}$, where t = Week)
      If yes: Week = Week - 1 (Go to 6.5)
   6.7) Maximum overtime exceeded? ($WL_{st} + p_j > Q_{st} + MO_{st}$, where t = Week)
      If yes: Week = Week - 1 (Go to 6.5)
      Otherwise: x_j = Week (Go to 6.8)
   6.8) Update total workloads ($WL_{st} = WL_{st} + p_j$)
7) For all operations within $J_{i'}$ that require production at s ($y_j = s$):
   CFL Approach

8) For all departments $s \in S$:
9) For all operations within $J_{i'}$ that require production at s ($y_j = s$):
   CFL Approach

10) Decisions extra nonregular capacity to avoid tardiness

Figure B.3; HL approach procedure

UFL approach

1) For all orders $i \in N$ do the following:
2) Select order with earliest due date (ad_i or $cd_{i'}$)
    - 2.1) Orders with equal due dates?
        - Select order with shortest total processing time (workload)
    - 2.2) Orders with equal due dates and equal workloads?
        - Select first order within N
3) Does selected order have a due date within a frozen period? ($ad_i \leq |H|$)
    - If yes: Do not change loading periods of order ($x_j = z_j \quad \forall j \in J_i$)
      Update total workloads ($WL_{st} = WL_{st} + p_j$) & Go to 2
    - Otherwise: Go to 4
4) For all departments $s \in S$:
5) For all operations within J_i that require production at s ($y_j = s$):
    - 5.1) Select operation with earliest release date
        - 5.1.1) Operations with equal release dates?
            - Select operation with shortest processing time
        - 5.1.2) Operations with equal release dates and equal processing times?
            - Select first operation within $J_{i'}$
    - 5.2) Week $= r_j$
    - 5.3) No regular capacity available in Week? ($WL_{st} < Q_{st}$, where $t =$ Week)
        - If yes: Week = Week + 1 (Go to 5.3)
    - 5.4) Maximum overtime exceeded? ($WL_{st} + p_j > Q_{st} + MO_{st}$, where $t =$ Week)
        - If yes: Week = Week + 1 (Go to 5.3)
        - Otherwise: $x_j =$ Week (Go to 5.5)
    - 5.5) Update total workloads ($WL_{st} = WL_{st} + p_j$)
6) Decisions extra nonregular capacity to avoid tardiness

Figure B.4; UFL approach procedure

UCL approach

1) For all orders $i \in N$ do the following:
2) Select order with earliest due date (ad_i or $cd_{i'}$)
   2.1) Orders with equal due dates?
      Select order with shortest total processing time (workload)
   2.2) Orders with equal due dates and equal workloads?
      Select first order within N
3) Does selected order have a due date within a frozen period? ($ad_i \leq |H|$)
   If yes: Do not change loading periods ($x_j = z_j \quad \forall j \in J_i$)
   Update total workloads ($WL_{st} = WL_{st} + p_j$) & Go to 2
   Otherwise: Go to 4
4) For all departments $s \in S$:
5) For all operations within J_i that require production at s ($y_j = s$):
   5.1) Determine collective loading period ($\max_{j \in J_{i'} | y_j = s} \{r_j\}$)
   5.2) Select operation with earliest release date
      5.2.1) Operations with equal release dates?
         Select operation with shortest processing time
      5.2.2) Operations with equal release dates and equal processing times?
         Select first operation within $J_{i'}$
   5.3) Week = Collective loading period
   5.4) No regular capacity available in Week? ($WL_{st} > Q_{st}$, where t = Week)
      If yes: Week = Week + 1 (Go to 5.4)
   5.5) Maximum overtime exceeded? ($WL_{st} + p_j > Q_{st} + MO_{st}$, where t = Week)
      If yes: Week = Week + 1 (Go to 5.4)
      Otherwise: x_j = Week (Go to 5.6)
   5.6) Update total workloads ($WL_{st} = WL_{st} + p_j$)
6) Decisions extra nonregular capacity to avoid tardiness

Figure B.5; UCL approach procedure

UHL approach

1) For all orders $i \in N$ do the following:
2) Select order with earliest due date (ad_i or $cd_{i'}$)
    2.1) Orders with equal due dates?
        Select order with shortest total processing time (workload)
    2.2) Orders with equal due dates and equal workloads?
        Select first order within N
3) Is selected order the incoming order i'?
    If yes: Go to 4
    Otherwise Go to 5
4) Incoming order requests compliance with due date? ($RCE_{i'} > 0$)
    If yes: Go to 6
    Otherwise Go to 10
5) Does selected order have a due date within a frozen period? ($ad_i \leq |H|$)
    If yes: Do not change loading periods ($x_j = z_j \quad \forall j \in J_i$)
        Update total workloads ($WL_{st} = WL_{st} + p_j$) & Go to 2
    Otherwise: Go to 6
6) For all departments $s \in S$:
7) Does department desires early completion? ($ACE_s > 0$)
    If yes: Go to 8
    Otherwise: Go to 9
8) For all operations within J_i that require production at s ($y_j = s$):
    8.1) Select operation with latest release date
        8.1.1) Operations with equal release dates?
            Select operation with shortest processing time
        8.1.2) Operations with equal release dates and equal processing times?
            Select first operation within $J_{i'}$
    8.2) Counter = 0
    8.3) Week = $cd_{i'} - q$ or $ad_i - q$
    8.4) Counter = Counter + 1
    8.5) $r_j >$ Week?
        If yes: Week = $cd_{i'} - q$ + Counter or $ad_i - q$ + Counter (Go to 8.4)
        Otherwise: Go to 8.6
    8.6) No regular capacity available in Week? ($WL_{st} > Q_{st}$, where t = Week)
        If yes: Week = Week - 1 (Go to 8.5)
    8.7) Maximum overtime exceeded? ($WL_{st} + p_j > Q_{st} + MO_{st}$, where t = Week)
        If yes: Week = Week - 1 (Go to 8.5)
        Otherwise: x_j = Week (Go to 8.7)
    8.7) Update total workloads ($WL_{st} = WL_{st} + p_j$)
9) For all operations within J_i that require production at s ($y_j = s$):
    CFL Approach
10) For all departments $s \in S$:
11) For all operations within $J_{i'}$ that require production at s ($y_j = s$):
    CFL Approach
12) Decisions extra nonregular capacity to avoid tardiness

Figure B.6; UHL approach procedure

Prevent tardiness approach

1) Best solution = Initial solution
   Current solution = Initial solution
2) Does order have tardiness? ($d_{i'} > cd_{i'}$ or $d_i > ad_i$)
   If yes: Go to 3
   Otherwise: Go to 7
3) Determine all bottleneck operations of the order in the current solution ($j|x_j = \max_s\{c_{is}\}$)
4) For all bottleneck operations, do the following:
   4.1) Select bottleneck operation with latest release date
      4.1.1) Operations with equal release dates?
         Select operation with shortest processing time
      4.1.2) Operations with equal release dates and equal processing times?
         Select first operation within J_i
   4.2) Release date of operation equal to loading period? ($r_j = x_j$)
      If yes: Go to 7
   4.3) Load operation 1 week earlier ($x_j = x_j - 1$)
   4.4) Maximum nonregular capacity exceeded? ($WL_{st} + p_j > Q_{st} + MO_{st}$)
      If yes: Go to 4.2
   4.5) Update total workloads ($WL_{st} = WL_{st} + p_j$)
5) Obtain new solution
6) Objective new solution < Best solution
   If yes: Current solution = New solution
   Best solution = New solution (Go to 2)
   Otherwise: Current solution = New solution (Go to 2)
7) Result = Best solution

Figure B.7; Procedure decision regarding extra overtime usage to prevent tardiness of order

Appendix C: Model parameters

This Appendix elaborates on how we acquired the model parameters that together determine the objective function value in the setting of TOMRA. We derived the parameter values based on several interviews and other verbal communication methods with the production managers at TOMRA.

First of all, the managers at TOMRA indicated that they rather desire a spread of the overtime hours instead of having large fluctuations per week. Moreover, they mentioned that, for instance, the tenth overtime hour in a week should be penalized more than the first one. All in all, the way that they desire this results in an exponent of the overtime costs of 2 (Exp = 2).

Additionally, if the regular capacity in a week is currently fully occupied without overtime hours, TOMRA is willing to assign at most five overtime hours per employee per week to prevent tardiness on an incoming order. This implies that if we set CO equal to 1, then RCT would equal 25 (resulting from the model described in Section 4.3).

TOMRA considers existing orders with already assigned due dates twice as important as incoming orders, resulting that ACT becomes twice the value of RCT, which is 50. Moreover, TOMRA does not desire finishing early with the production activities at the table and conveyor departments (s = 1 & 3). However, they of course rather desire earliness than tardiness. In our setting, ACE_1 and ACE_3 are assigned a value of 3 after close contact with the managers, whereas ACE_2 equals 0.

The costs regarding earliness on the customer-requested due date equals 10 for the incoming orders in our experimental design. In reality, this value can different per customer, as sometimes customers do desire a due date as soon as possible. However, a value of $RCE_{i'} \leq 0$ would imply that the CFL & HL and UCL & UHL are exactly the same. Therefore, we assume that the incoming order does not desire an early due date in our setting.

At last, the spread costs SC for spreading the workload of an order at a department over multiple weeks is assigned a value of 5. The reasoning behind this is that TOMRA indicated that they would rather spread the workload over multiple weeks than imposing tardiness on the customer-requested due date of an incoming order.

Table B.1 highlights the values in the setting of TOMRA of all discussed model parameters.

Table B.1; Model parameters values in our setting

Model parameter	Value
Exp	2
$RCE_{i'}$	10
RCT	25
ACE_1	3
ACE_2	0
ACE_3	3
ACT	50
SC	5
CO	1

Appendix D: Experiment instances

This Appendix describes the problem instances in the setting of TOMRA. These instances consist of existing orders that occupy capacity at the moment of the arrival of an incoming order. In total, we consider 4 different instances. For each of these instances, we indicate the total number of orders as well as the peak demand period where the customer requests a due date for the incoming order.

For every instance, we considered the (past) employee availabilities at TOMRA per production department for the entire time horizon to obtain the production capacity. Here, we made some estimations for weeks where there was no full information regarding the number of employees that were available. At TOMRA, every FTE works 38 hours per week (to obtain Q_{st}) and we allow a maximum of 10 overtime hours per FTE in a week (for MO_{st}).

Instance 1

The first situation regards the existing orders at 01/05/2022. In total, we consider 332 existing orders within the time horizon of 20 weeks, making the set N regarding all orders (including the incoming order) as follows: $N = \{1, \dots, 333\}$.

The first situation has few or none available capacity in the first three weeks at all production departments. Because of this and since we consider a slack q of 1 week in our setting, we set the customer-requested due date ($cd_{i'}$) equal to 4.

Instance 2

The second problem instance considers existing orders at 14/11/2022. In total, these are 309 existing orders within the stated time horizon of 20 weeks, making the set making the set N as follows: $N = \{1, \dots, 310\}$. Moreover, the second situation has few or none available capacity in the first five weeks at all production departments. Because of this, we set the customer-requested due date ($cd_{i'}$) equal to 6.

Instance 3

The third instance considers existing orders at 23/05/2022. In total, these are 290 existing orders within the stated time horizon of 20 weeks, making the set making the set N as follows: $N = \{1, \dots, 291\}$. The third situation has few or none available capacity in the first four weeks at all production departments. Because of this, we set the customer-requested due date ($cd_{i'}$) equal to 5.

Instance 4

The fourth instance considers existing orders at 16/01/2022. In total, these are 432 existing orders within the stated time horizon of 20 weeks, making the set making the set N as follows: $N = \{1, \dots, 433\}$. The fourth situation has few or none available capacity in the first three weeks at most production departments. Because of this, we set the customer-requested due date ($cd_{i'}$) equal to 4.

Table D.1 presents an overview of the information per experiment instance.

Table D.1; Overview of experiment instances

Instance	**1**	**2**	**3**	**4**
Existing orders	332	309	290	432
Requested due date ($cd_{i'}$)	4	6	5	4

Appendix E: Incoming order experiments

This Appendix elaborates in detail on the different incoming orders that we consider and evaluate the solution approaches on. In total, we consider 4 different incoming orders; one relatively small order and one larger order at all departments and two orders that require production activities at two departments.

Incoming order 1

The small incoming order at all departments comes from a hypothetical Dutch customer (impacts some release dates, see Appendix F). This incoming order requires products from all three production departments and consists of a total of 11 operations:

- 4 operations at the table department ($s = 1$) with a total workload of 10 hours.
  - 4 small straight MUP2 tables
- 6 operations at the roller belt department ($s = 2$) with a total workload of 8.5 hours.
  - 2 elevator belts start/stop 2 meter
  - 1 PET elevator belt (470mm)
  - 1 ECO-WALL T-9 300
  - 2 V2 conveyors
- 1 operation at the conveyor department ($s = 3$) with a total workload of 10 hours
  - 1 medium Z conveyor

Incoming order 2

The large incoming order at all departments comes from a hypothetical foreign customer. This incoming order also requires products from all three production departments and consists of a total of 29 operations:

- 10 operations at the table department ($s = 1$) with a total workload of 40 hours.
  - 10 extra-large straight MUP2 tables
- 15 operations at the roller belt department ($s = 2$) with a total workload of 30 hours.
  - 4 elevator belts 2 meter
  - 2 PET elevator belt (160mm)
  - 1 ECO-WALL T-9/T-8 600
  - 4 Mergers
  - 4 V2 conveyors
- 4 operations at the conveyor department ($s = 3$) with a total workload of 40 hours
  - 2 large pre-feeders
  - 2 medium depot conveyors

Incoming order 3

The incoming order that requires production activities at departments 1 and 2 comes from a hypothetical Dutch customer and consist of a total of 12 operations:

- 6 operations at the table department ($s = 1$) with a total workload of 14 hours.
  - 6 medium straight EP tables
- 6 operations at the roller belt department ($s = 2$) with a total workload of 13 hours.
  - 3 elevator belts start/stop 1.5 meter
  - 1 PET elevator belt (160mm)
  - 2 V2 conveyors

Incoming order 4

The incoming order that requires production activities at departments 2 and 3 comes from a hypothetical foreign customer and consist of a total of 9 operations:

- 6 operations at the roller belt department ($s = 2$) with a total workload of 9 hours.
  - 3 PET elevator belt (470mm)
  - 3 Mergers
- 3 operations at the conveyor department ($s = 3$) with a total workload of 22 hours.
  - 2 small pre-feeders
  - 1 Wash & Waste

Like in Appendix C, we mention that all incoming orders do not desire an early due date ($RCE_{i'} > 0$), where $RCE_{i'}$ equals 10.

Table E.1 presents a summary of the different incoming orders for our experimental design.

Table E.1; Characteristics of incoming orders

Incoming order		**1**	**2**	**3**	**4**
Department $s = 1$	Workload (hours)	10	40	14	*X*
	Operations	4	10	6	*X*
Department $s = 2$	Workload (hours)	8.5	30	13	9
	Operations	6	15	6	6
Department $s = 3$	Workload (hours)	10	40	*X*	22
	Operations	1	4	*X*	3
Total workload (hours)		28.5	110	27	31
Total operations		11	29	12	9

Appendix F: Release dates determination

This Appendix outlines how we determine release dates of the considered products in the setting of TOMRA. These release dates are part of the model input and form an essential part of the solution approaches from Chapter 5.

For all the products that are being produced at the table department and conveyor department, the release dates depend on whether the products are from Dutch orders or foreign orders as mentioned in Section 2.2.2. The Dutch orders have checks on the tables and conveyors approximately one week before their assigned due date. However, since an incoming order does not have an assigned due date but a speculative requested due date, its release dates for the products at the table and conveyor departments depend on the earliest possible moment when these checks can be executed. In practice, one should consider the available capacity of the employees that execute the design checks at the location of the customer. For this research, we make an assumption related to this, namely that the release dates for products at the table and conveyor department of an incoming Dutch orders are two weeks after the moment of the incoming order request.

Furthermore, foreign orders that arrive at TOMRA Apeldoorn by foreign TOMRA departments do not have restrictions on their release dates. This implies that tables and conveyors for both foreign existing and incoming orders can immediately start their production.

Table D.1 proves an overview of the release dates for products that are being produced at the table and conveyor departments of Dutch and foreign existing and incoming orders.

Table D.1; Release dates for products at the table and conveyor departments

	Type of order	**Release date**
Dutch orders	Existing orders	$r_j = ad_i - 1$
	Incoming orders	$r_j = 2$
Foreign orders	Existing orders	$r_j = 0$
	Incoming orders	$r_j = 0$

Note that a slack for determining the due dates larger than one ($q > 1$) would automatically result in tardiness related to already assigned due dates of existing Dutch orders.

For the standardizes products that are being produced roller belt department, TOMRA always knows the exact number and specification within a Dutch or foreign order at the moment of its arrival. The release dates of these products depend in the material availabilities.

In total, we consider 12 materials with stochastic external lead times. We chose this selection based on recommendations and interviews with production managers and employees. Table D.2 highlights these materials as well as for which products they are necessary and their quoted deterministic lead time in weeks.

All other products, all materials are available at the moment of the arrival (reasonable as told)

Table D.2: Details on materials with stochastic external lead time

Material	Required for	Quoted deterministic lead time (weeks)
A	Elevator belts, Elevator belts Start/Stop, PET elevator belts and Mergers	7
B	Elevator belts, Elevator belts Start/Stop, PET elevator belts and Mergers	7
C	Elevator belts, Elevator belts Start/Stop, PET elevator belts and Mergers	7
D	Elevator belt, Elevator belts Start/Stop, PET elevator belts and Mergers	7
E	Mergers	7
F	V2 Conveyor	7
G	V2 Conveyor	7
H	V2 Conveyor	7
I	V2 Conveyor	7
J	V2 Conveyor	7
K	V2 Conveyor	7
L	V2 Conveyor	7

Next, we provide an overview of the availabilities of these materials at the moment of a new order arrival per experimental instance and number of periods passed since last replenishment order from raw material l (RO_l). We assume that the upcoming replenishments orders for the materials are sufficient to realize the production of all requested orders elements for all 4 incoming orders. Table D.3 contains this information per experiment.

Table D.3; Material availabilities per instance

Material	Instance 1		Instance 2		Instance 3		Instance 4	
	Available?	RO_l	Available?	RO_l	Available?	RO_l	Available?	RO_l
A	No	5	Yes	4	Yes	4	No	4
B	No	5	No	3	No	6	Yes	5
C	Yes	6	Yes	4	No	5	Yes	6
D	Yes	5	No	5	Yes	5	No	4
E	No	4	No	3	No	6	Yes	4
F	No	6	Yes	3	Yes	4	No	5
G	Yes	4	No	4	No	4	No	4
H	Yes	5	No	4	Yes	3	Yes	3
I	No	4	Yes	5	No	4	No	5
J	Yes	6	Yes	3	Yes	5	No	6
K	Yes	6	No	4	No	6	Yes	4
L	No	5	Yes	4	Yes	3	Yes	5

For determining the release dates, we use the SAA approach described in Section 4.4. More specifically, we first determine release dates of operations that require one or multiple materials with stochastic external lead times based on a certain percentile to construct a solution and afterwards randomly select scenarios to assess the feasibility and obtain an estimate of the service level target λ.

Per instance, we select external lead times based on the 75th percentile. Moreover, to be consistent in comparing the approaches, we select the same random scenarios for each approach per problem instance when assessing the solution feasibility for an instance. In total, we assess the feasibility of a solution for 25 scenarios.

Appendix G: Improvement heuristics parameters

This Appendix outlines the parameters that we apply for both improvement heuristics from Section 5.2. We tune these parameters by assessing the performance of certain parameter values in terms of objective function value and computational time. We elaborate in detail on the process of the parameter tuning for both the adapted steepest descent and simulated annealing heuristics.

Adapted steepest descent parameters

For the adapted steepest descent heuristic (ASD), the goal is to find better neighborhood solution in a timely manner. For this improvement heuristic, we have two parameters, namely the Max Iterations and the Time Limit. The challenge for tuning these parameters is that each of the construction heuristic from Section 5.1 performs in a different manner and can each desire different parameter settings. Because of this, we perform experiments on certain values of the Max Iterations as well as on several values for the Time Limit on each construction heuristic. For these experiments, we use instance 1 (see Appendix D) with incoming order 1 (see Appendix E). For the Max Iterations, we experiment on values of 50, 100, 150, 200 and 250, where we set the Time Limit to 300 seconds (5 minutes) to see which improvements can take place within 300 seconds.

For all combinations of parameters settings and construction heuristics, we perform 3 replications to obtain more generalized results. The results can be seen in Tables G.1 through G.3, where we indicate the average objective and average time (seconds) of each combination with standard deviations noted in brackets.

Table G.1; Experiment results on adapted steepest descent parameters for FL and CFL approaches

		FL Approach		CFL Approach	
Max Iterations	**Time Limit**	**Objective**	**Time**	**Objective**	**Time**
50	300	356.9 (11.1)	17.7 (6.6)	342.2 (2.4)	16.2 (5.8)
100	300	357.8 (6.0)	33.7 (10.8)	342.4 (0.4)	30.2 (7.4)
150	300	359.2 (4.9)	70.6 (11.8)	340.4 (1.8)	77.1 (25.1)
200	300	350.8 (8.5)	155.6 (4.3)	338.5 (3.4)	173.7 (32.8)
250	300	348.5 (10.3)	248.8 (17.3)	333.1 (1.7)	181.1 (72.7)

Table G.1; Experiment results on adapted steepest descent parameters for HL and UFL approaches

		HL Approach		UFL Approach	
Max Iterations	**Time Limit**	**Objective**	**Time**	**Objective**	**Time**
50	300	336.1 (1.7)	9.4 (3.3)	842.6 (14.9)	300.0 (0.0)
100	300	337.1 (0.0)	19.9 (11.4)	951.4 (22.1)	300.0 (0.0)
150	300	336.1 (1.7)	96.7 (21.9)	1002.14 (5.7)	300.0 (0.0)
200	300	330.1 (6.2)	165.8 (48.5)	1057.4 (17.4)	300.0 (0.0)
250	300	328.1 (3.0)	229.4 (37.8)	1070.1 (23.9)	300.0 (0.0)

Table G.3; Experiment results on adapted steepest descent parameters for UCL and UHL approaches

		UCL Approach		UHL Approach	
Max Iterations	**Time Limit**	**Objective**	**Time**	**Objective**	**Time**
50	300	812.0 (23.6)	300.0 (0.0)	474.2 (0.0)	5.7 (0.0)
100	300	901.16 (8.6)	300.0 (0.0)	432.5 (33.8)	53.1 (21.1)
150	300	936.3 (10.3)	300.0 (0.0)	398.2 (26.2)	116.3 (51.3)
200	300	982.9 (11.7)	300.0 (0.0)	406.8 (54.5)	138.0 (44.3)
250	300	990.1 (16.9)	300.0 (0.0)	384.9 (8.7)	195.4 (20.1)

From the results of the tuning experiments, we observe that for both the UFL and UCL construction heuristics, the time limit of 300 seconds is reached for all five levels of Max Iterations. This implies that there is still improvement possible for these situations after 300 seconds. However, we observe that for these two construction heuristics 50 Max Iterations result in the lowest objective values within the Time Limit. Additionally, for the remaining construction heuristics (FL, CFL, HL and UHL), the Time Limit is never reached on average, meaning that no further improvements were found resulting in the ASD procedure to stop. Because of this, the highest number of Max Iterations, which is 250, results in the lowest objective values for these four construction heuristics.

We choose for construction heuristics UFL and UCL to set the Max Iterations to 50. Additionally, for the FL, CFL, HL and UHL construction heuristics, we set the Max Iterations to 250.

The main reason for not choosing a higher value of Max Iterations for the UFL and UCL construction heuristics is that with 50 Max Iterations, the objective values converge faster and result in more favorable values. Needless to say, would the Time Limit be extended, then more Max Iterations would probably result in lower objective values. However, since our aim is to find good solutions in a timely manner, we set the Max Iterations for these 2 construction heuristics to 50. We do not try lower values of Max Iterations for these approaches, as we feel that this can result in overfitting to the specific problem instance (instance 1 with incoming order 1). For the same reasoning, we do not select higher values of Max Iterations for the FL, CFL, HL and UHL construction heuristic.

Simulated annealing parameters

The first step in tuning the parameters for the SA improvement heuristic is to determine the starting temperature. We obtain this temperature by solving instance 1 (see Appendix D) with incoming order 1 (see Appendix E) with a starting temperature of 150, where the FL approach constructs the initial solution. Moreover, we solve this instance with a Markov chain length of 200 and a cooling factor α of 0.99. At the end of each Markov chain, we store the acceptance ratio that indicates the number of accepted neighbors divided by the generated neighbors of the corresponding Markov chain. Figure G.1 provides the results of this procedure.

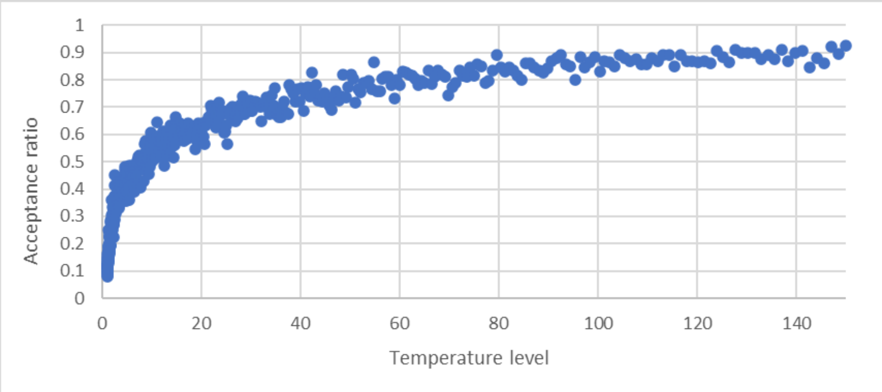

Figure G.1; Neighborhood acceptance ratio for various temperature levels

For our approach, we choose to use a starting temperature of 30, which has an acceptance ratio of about 0.70. We choose not to start the SA heuristic with a higher temperature level, since each of the construction heuristics from Section 5.1 have logic about their way of creating initial solutions and do not construct random solutions for instance. Because of this, choosing a higher starting temperature level can make initial solutions significantly worse at the start of the SA heuristic, which can result in computational waste.

Afterwards, we consider stop temperature levels of 2.5, 5, and 7.5, Markov chain lengths of 100, 150 and 200 and cooling factors (α) of 0.925, 0.950 and 0.975. For each combination of parameters, we conduct 3 replications with again solving instance 1 with incoming order 1 with FL as the construction heuristic to diminish randomness, which enables us to obtain more generalized results. As in theory, the SA heuristic is capable of finding the global optima from every initial solution, we only use one construction heuristic (FL in our case) in tuning the SA parameters.

Additionally, we do not consider lower stop temperature levels, higher Markov chain lengths of higher cooling factors as the computational times limit us in exploring the performances on these computationally expensive parameter settings. Moreover, for selecting the parameters of the SA heuristic, we do not consider a time limit like with the ASD heuristic. The reason for this is that the SA heuristic can escape from local optima, which can take more time. Eventually, we apply the SA heuristic to our problem setting to see how much improvement can be achieved, where there obviously is a trade-off between computational time and objective value.

Table G.4 highlights the results of the experiments for tuning the SA parameters, where we provide the average solution value and objective function as well as the standard deviations noted in brackets.

Table G.4; Experiment results for tuning the simulated annealing parameters

Scenario	Start Temp	Stop Temp	MC length	Alpha	Objective	Time
1	30	7.5	100	0.925	360.3 (2.5)	226.4 (16.4)
2	30	7.5	100	0.95	355.4 (1.6)	382.4 (5.7)
3	30	7.5	100	0.975	350.7 (8.7)	741.8 (14.0)
4	30	7.5	150	0.925	350.1 (12.1)	357.9 (4.6)
5	30	7.5	150	0.95	353.8 (2.9)	563.4 (5.8)
6	30	7.5	150	0.975	342.5 (8.4)	1125.6 (19.9)
7	30	7.5	200	0.925	347.5 (2.9)	480.8 (12.5)
8	30	7.5	200	0.95	343.8 (3.1)	740.8 (9.2)
9	30	7.5	200	0.975	341.8 (1.2)	1432.0 (20.3)
10	30	5	100	0.925	356.7 (4.1)	288.4 (17.7)
11	30	5	100	0.95	353.8 (4.2)	477.0 (7.1)
12	30	5	100	0.975	347.5 (6.6)	963.7 (25.0)
13	30	5	150	0.925	348.5 (10.7)	458.4 (3.5)
14	30	5	150	0.95	349.7 (2.9)	704.4 (5.8)
15	30	5	150	0.975	337.1 (11.4)	1454.7 (24.9)
16	30	5	200	0.925	345.8 (2.9)	615.1 (12.6)
17	30	5	200	0.95	343.8 (3.1)	929.0 (8.1)
18	30	5	200	0.975	336.5 (2.1)	1838.4 (40.4)
19	30	2.5	100	0.925	352.1 (6.9)	397.1 (18.3)
20	30	2.5	100	0.95	351.1 (6.6)	665.8 (13.3)
21	30	2.5	100	0.975	340.4 (2.8)	1341.9 (34.5)
22	30	2.5	150	0.925	345.8 (8.4)	638.5 (3.2)
23	30	2.5	150	0.95	341.8 (1.1)	988.8 (3.8)
24	30	2.5	150	0.975	333.5 (9.5)	2033.6 (35.2)
25	30	2.5	200	0.925	344.8 (4.0)	854.8 (9.7)
26	30	2.5	200	0.95	341.8 (2.1)	1302.0 (8.2)
27	30	2.5	200	0.975	331.5 (0.6)	2534.1 (21.8)

Figure G.2 depicts the results of the experiments. We choose to select the parameter values from the 15th scenario for running the simulated annealing heuristic. The reasoning behind this is that this scenario results in the best trade-off between computational time and objective value in our eyes, as we aim to find good quality solutions in a timely manner.

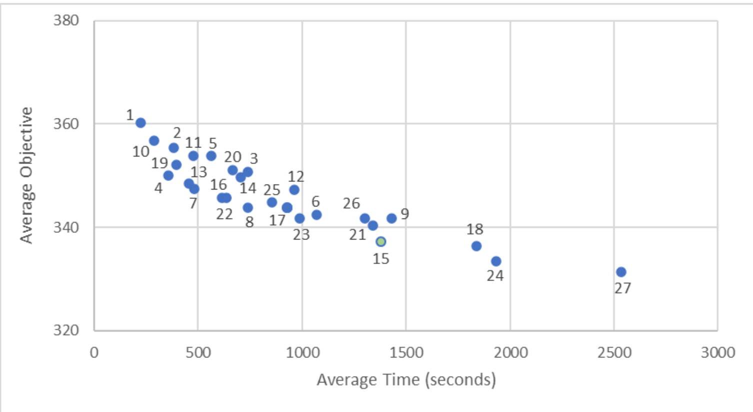

Figure G.2; Depicted simulated annealing parameter experiment results